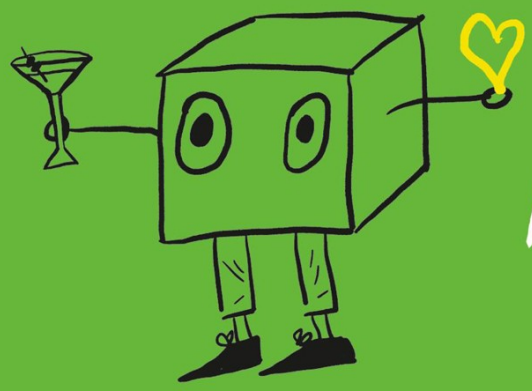
How **Artificial Intelligence** Works and Why It's Making the World a Weirder Place

YOU LOOK LIKE A THING AND I LOVE YOU



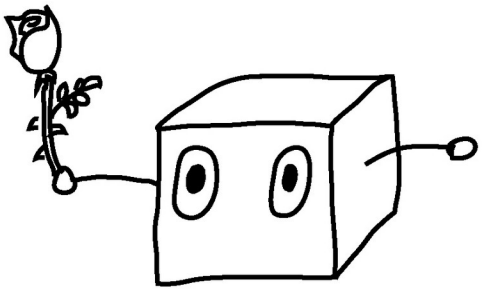
Janelle Shane

"I can't think of a better way to learn about artificial intelligence, and I've never had so much fun along the way." —ADAM GRANT, *New York Times* bestselling author of *Originals*

ДЖАНЕЛЬ ШЕЙН

Ти схожа на якусь штуку, і я тебе люблю

Як працює штучний інтелект і чому він робить світ дивнішим



Повний український переклад

Збережено структуру, абзаци й оригінальні ілюстрації; до змісту додано тематичні мініглави.

YOU LOOK LIKE A THING AND I LOVE YOU

How Artificial Intelligence Works
and Why It's Making the World
a Weirder Place

Janelle Shane

VORACIOUS

Little, Brown and Company
New York ▪ Boston ▪ London

Оригінальний титульний аркуш

Авторські права

Авторські права © 2019 Джанель Шейн

Дизайн обкладинки — Капо Нг; ілюстрація на обкладинці — Капо Нг і Джанель Шейн

Авторські права на обкладинку © 2019 Hachette Book Group, Inc.

Hachette Book Group підтримує право на свободу вираження поглядів і цінність авторського права. Мета авторського права — заохочувати письменників і митців створювати твори, що збагачують нашу культуру.

Сканування, завантаження й поширення цієї книги без дозволу є викраденням інтелектуальної власності авторки. Щоб отримати дозвіл на використання матеріалів книги — за винятком цитування в оглядах, — зверніться за адресою permissions@hbgusa.com. Дякуємо за підтримку прав авторки.

Voracious / Little, Brown and Company

Hachette Book Group

1290 Avenue of the Americas, New York, NY 10104

littlebrown.com

twitter.com/littlebrown

facebook.com/littlebrownandcompany

Перше електронне видання: листопад 2019 року

Voracious — імпринт Little, Brown and Company, підрозділу Hachette Book Group, Inc. Назва й логотип Voracious є торговельними марками Hachette Book Group, Inc.

Видавець не відповідає за сайти та їхній вміст, якщо ці сайти не належать видавцеві.

Hachette Speakers Bureau представляє широке коло авторів для участі в публічних заходах. Докладніше: hachettespeakersbureau.com або за телефоном (866) 376-6591.

Усі зображення створила авторка, крім таких: зображення GANcats, надане NVIDIA Corporation за ліцензією Creative Commons BY-NC 4.0 і використане з дозволу; зображення кенгуру Quick, Draw!, надані Google за ліцензією Creative Commons BY 4.0; зображення планів школи © Джоел Саймон, використані з дозволу; зображення субмарин © Денні Кармон, Йоав Ґолдберґ і Деніел Зоран, використані з дозволу; зображення лижників © Ендрю Ільяс, Логан Енгстром, Аніш Аталіє та Джессі Лін, використане з дозволу.

ISBN 978-0-316-52523-7

E3-20190927-JV-NF-ORI

Зміст

Назви мініглав описують зміст і можуть містити спойлери.

Авторські права
Присвята
ВСТУП: ШІ вже всюди
Як ШІ навчався фліртувати
ШІ вже всюди — але він не безпомилковий
Коли помилки алгоритмів стають небезпечними
Експерименти AI Weirdness як спосіб зрозуміти ШІ
П’ять принципів дивакуватості ШІ
Що ми з’ясуємо в цій книзі
РОЗДІЛ 1: Що таке ШІ?
Чому назвою «ШІ» позначають різні речі
СТУК-СТУК, ХТО ТАМ?
Програми з правилами проти машинного навчання
Програмування за правилами: жарти з готового шаблону
Навчання ШІ: від випадкових літер до структури жарту
Перший справжній жарт, створений ШІ
Один метод — нові птахи, парфуми й рецепти
ПРОСТО ДОЗВОЛЬТЕ ШІ САМОМУ РОЗІБРАТИСЯ
Як ШІ знаходить правила, яких ми не знаємо
Гібридні програми та псевдо-ШІ
Шахи, ігри та несподівані лазівки
ІНОДІ ЙОГО ПРАВИЛА ПОГАНІ
«Зелена трава дорівнює вівці»
Система виявлення раку, яка насправді шукала лінійки
ЯК ВИЯВИТИ ПОГАНЕ ПРАВИЛО
Чому внутрішні правила ШІ непрозорі
Як ШІ копіює людські упередження
ЧОТИРИ ОЗНАКИ МАЙБУТНЬОЇ КАТАСТРОФИ ШІ
Як насправді може початися катастрофа ШІ
Ознака № 1: завдання надто складне
Ознака № 2: справжнє завдання не таке, як ми думали
Ознака № 3: існують підступні короткі шляхи
Ознака № 4: ШІ навчався на недосконалих даних
КАТАСТРОФА — ЧИ ЗАХВАТ
Успіх залежить від придатності завдання для ШІ
РОЗДІЛ 2: ШІ всюди, але де саме?
ЦЕЙ ПРИКЛАД СПРАВЖНІЙ, Я НЕ ЖАРТУЮ
Ферма, що вирощує шість мільярдів тарганів
Чому ШІ добре підходить для оптимізації ферми
Коли оптимізація відчиняє тарганам двері
ВЛАСНЕ, Я НЕ ПРОТИ, ЩОБ РОБОТ ЗАБРАВ ЦЮ РОБОТУ
Автоматизація масових і повторюваних завдань
Медичні зображення, спам і гіперперсоналізація
Новини з таблиці: журналістський алгоритм Heliograf
Астрономія, творчі інструменти та дипфейки
Послідовність не означає неупередженість
ЩО ВУЖЧЕ ЗАВДАННЯ, ТО РОЗУМНІШИМ ЗДАЄТЬСЯ ШІ

Тест Тюрінга можна обійти вузькою темою
Як людська ввічливість допомагає чатботам
Чому Facebook M не став універсальним помічником
Вузьке й широке навчання на рецептах
С-ЗРО ПРОТИ ВАШОГО ТОСТЕРА
Вузький ШІ та загальний штучний інтелект
Чому розпізнавання емоцій не є вузьким завданням
НЕДОСТАТНЬО ДАНИХ — ОБЧИСЛЕННЯ НЕМОЖЛИВЕ
ШІ потребує тисяч і мільйонів прикладів
Прискорене навчання в симуляціях
Ризик навчання у спрощеному світі
ЯК СКОРИСТАТИСЯ ЧУЖИМ ПРОГРЕСОМ
Перенесення навчання: від метал-гуртів до морозива
Повторне використання моделей розпізнавання зображень
НЕ ПРОСІТЬ ЙОГО ПАМ’ЯТАТИ
Короткозорий ШІ у відеоіграх
Чому алгоритм не може злізти з уступу
Коротка пам’ять у перекладі й створенні тексту
Шістдесят п’ять символів пам’яті й кривавий пудинг
GPT-2 довше утримує тему
Як групування одиниць допомагає пам’яті
ЧИ Є ПРОСТІШИЙ СПОСІБ РОЗВ’ЯЗАТИ ЦЮ ПРОБЛЕМУ?
Коли ШІ справді є найпростішим рішенням
Коли замість ШІ достатньо здорового глузду
ДОЗВОЛИТИ ШІ КЕРМУВАТИ?
Чому безпілотне керування так приваблює
Рідкісні втручання людини, яких не вдається позбутися
Фатальні помилки розпізнавання перешкод
Кенгуру, ему, лава й нескінченність рідкісних ситуацій
Контрольовані маршрути наближають нас до громадського транспорту
Проблема автоматизації третього рівня
Безпілотні автомобілі потребують компромісів або гнучкішого ШІ
РОЗДІЛ 3: Як він насправді навчається?
Як випадкові літери стають жартами
НЕЙРОННІ МЕРЕЖІ
Мозок потужністю в одного черв’яка
Чому людська кількість нейронів ще не дасть людського інтелекту
МАГІЧНА СЕНДВІЧНА ДІРА
Один чудовий сендвіч на тисячу жахливих
Чорна скринька з інгредієнтами на вході й смакотою на виході
Проста мережа додає ваги інгредієнтів
Помилка великого сендвіча: смакота може «скасувати» мульчу
Прихований шар навчається поєднувати ознаки
Клітина-каральник перемагає помилку великого сендвіча
Клакерфлафер карає саме курятину з маршмелоу
ПРОЦЕС НАВЧАННЯ
Навчання починається з випадкових і дуже поганих ваг
Тисячі ітерацій поступово виправляють мережу
Дисбаланс класів навчає ШІ відкидати взагалі все
КОЛИ КЛІТИНИ ПРАЦЮЮТЬ РАЗОМ
Більшість нейронів виконують незрозумілу частину спільної роботи
Нейрон настрою відрізняє похвалу від нищівної критики
На Едгарі По нейрон настрою помиляється
DeepDream змінює зображення, доки нейрони не побачать усюди собак
«Операція на мозку» генератора прибирає штори, але не столи
ЛАНЦЮГИ МАРКОВА

Швидке передбачення наступного слова з дуже короткою пам’яттю
Три слова пам’яті замикають пісні Disney в нескінченній петлі
Botnik лишає вибір дивних слів людині
Власні друкарські помилки можуть переслідувати вас місяцями
Інтернет навчив Google Docs виправляти правильні слова на неправильні
ВИПАДКОВІ ЛІСИ
Дерево рішень вирішує, чи евакуювати тарганячу ферму
Безліч слабких дерев голосує точніше за одне велике
ЕВОЛЮЦІЙНІ АЛГОРИТМИ
Підйом на пагорб може привести не на найвищу вершину
Еволюція перетворює «вгадай і перевір» на покоління рішень
Геном робота визначає форму тіла й рухи
Функція пристосованості й симуляція задають мету еволюції
Добір, мутація і схрещування створюють наступне покоління
Погана ціль еволюціонує робота, який убиває всіх
Зміна функції пристосованості перевиховує роботів-убивць
Найімовірніший ідеальний робот виявляється звичайними дверима
Еволюційні алгоритми проєктують бампери, білки й маховики
ГЕНЕРАТИВНО-ЗМАГАЛЬНІ МЕРЕЖІ (GAN)
Генератор і дискримінатор навчаються, обманюючи одне одного
GAN малює переконливих коней лише у дуже простому світі
Генератор перетворює випадковий шум на різних коней
Недосвідченого дискримінатора здатна обдурити ледь конеподібна пляма
GAN проводить тест Тюрінга, де сама є суддею й учасницею
Суперечливі цілі змушують GAN балансувати між наслідуванням та новизною
ЗМІШУВАННЯ, ПОЄДНАННЯ І СПІЛЬНА РОБОТА
Seeing AI об’єднує окремі моделі для опису світу вголос
Три алгоритми створюють, стилізують і називають картини
Зір, пам’ять і контролер разом ухиляються від вогняних куль у Doom
РОЗДІЛ 4: Він старається!
П’ять способів дати ШІ завдання, з яким він не впорається
НАДТО ШИРОКА ЗАДАЧА
StyleGAN малює реалістичних людей, але котів із зайвими очима
БІЛЬШЕ ДАНИХ, БУДЬ ЛАСКА
Вісім смаків морозива не пояснюють, що таке морозиво
Для ненавченої мережі літери, пробіли й переноси — лише числа
Фаза моторошної корови: мукання, шипіння й перші впізнавані слова
Ідеальне зазубрювання восьми прикладів не є творчим узагальненням
На 2011 прикладах мережа нарешті починає винаходити
Великі компанії вже мають дані, а решті доводиться їх збирати
Один «помічник» завантажує до набору весь каталог костюмів
Платний працівник може таємно доручити ботові навчати вашого бота
Один приклад можна розмножити дзеркаленням і заміною слів
Одне зображення показують сотні разів, щоб отримати мільйони діалогів
Після десяти запитань про жирафа люди переходять до квантової фізики
БЕЗЛАДНІ ДАНІ
Людські помилки в розмітці стають помилками безпілотного автомобіля
Рецепт наказує нарізати борошно кубиками й розмастити масло в холодильнику
Мережа чесно вивчає помилки, що вже були в рецептах
ДАНІ, ЩО МАРНУЮТЬ ЧАС
Половина мозку рецептурної мережі йде на вигадані ISBN та примітки
92 тисячі заголовків BuzzFeed змушують мережу одночасно вчити кілька мов
Котячі меми змушують GAN малювати псевдотекст і Grumpy Cat
BigGAN плутає футбольний м’яч із ногою, а мікрофон — із людиною
Очищення набору й обведення предметів допомагають алгоритмові
ЦЕ СПРАВЖНЄ ЖИТТЯ?
Через фотографічні звички людей ШІ бачить жирафів у кожному пейзажі

Непредставлені у Wikipedia науковиці стають невидимими й для ШІ
ІНШІ ДИВАЦТВА НАБОРІВ ДАНИХ
Нісенітниця сомалійською раптом перетворюється на біблійні пророцтва
Зміна пробілів змушує перекладач вигадувати форум та електронну адресу
Нейромережа може видати номери карток і соціального страхування
ВІДСУТНІ ДАНІ
Список котячих імен не містить слів, якими кота називати не варто
Коли дивність є метою, обмежене знання світу стає перевагою
Azure бачить у лінійному малюнку текстуру, якої там немає
Quick Draw знає лише 345 речей — зате мільйони їхніх ескізів
Автодоповнення правильно передбачає похорон бабусі — і саме тому помиляється
ТУТ ЧОТИРИ ЖИРАФИ
Якщо люди питають лише про видимі речі, ШІ навчається відповідати «так»
Запитання про кількість жирафів майже ніколи не має відповіді «нуль»
Блакитне яблуко для бота все одно червоне, а помаранчева вівця — біла
Коли бот не розуміє BB-8, він упевнено оголошує його собакою
Обмежена модель світу мусить вгадувати серед безмежної людської дивини
РОЗДІЛ 5: Чого ви насправді просите?
Чотири способи виконати завдання зовсім не так
Чому ШІ розв’язує не ту задачу
ШІ-демон навмисно шукає найлегшу лазівку
Робопси, які не хочуть ходити
Собаки й алгоритми вчаться не того, чого ми очікуємо
ЗЛАМУВАННЯ ФУНКЦІЇ ВИНАГОРОДИ
Дельфіни розмінюють сміття на рибу
Люди теж зламують стимули
Пожежі без заторів і школи без вікон
Поні з іменами, яких краще уникати
КОМП’ЮТЕРНІ ІГРИ ЗБИВАЮТЬ З ПАНТЕЛИКУ
«Перемогти» — надто нечітка мета
Tetris: поставити гру на вічну паузу
Mario не знає, що помирати погано
Вітрильник нескінченно збирає ті самі маркери
Oblivion: торговця вбили, а крамницю скупили
НЕ ХОДІТЬ
Навіщо ходити, якщо можна впасти?
Навіщо стрибати, якщо можна станцювати канкан?
Навіщо їхати, якщо можна крутитися?
Дурнуваті ходи: задом, боком і на одній нозі
Якщо сумніваєшся — не роби нічого
ДОПИТЛИВІСТЬ
Чому для допитливого ШІ смерть нудна
Допитливий робот-павук іде геть від фінішу
Venture: непередбачувані привиди цікавіші за перемогу
Гучний телевізор назавжди полонить ШІ
СТЕРЕЖІТЬСЯ ХИБНОЇ ФУНКЦІЇ ВИНАГОРОДИ
Від кумедних збоїв до серйозних наслідків
ШІ сховав супутниковий знімок усередині карти
YouTube: кліки породжують оманливі мініатюри
Час перегляду затягує людей у вир змов
Коли корпорація поводиться як ШІ з хибною метою
РОЗДІЛ 6: Злам Матриці, або ШІ знаходить спосіб
Футбольний м’яч розганяється до швидкості світла
Одноколісний велосипед відлітає в небеса
Люди будують Матриці, у яких навчаються ШІ
Кожна симуляція змушена спрощувати фізику

Дивна фізика робить безглузду ходу ефективною
НУ, ВИ Ж НЕ СКАЗАЛИ, ЩО ТАК НЕ МОЖНА
Нечіткі обмеження породжують технічно дозволені дивацтва
Оптичний проєкт працює завдяки лінзі завтовшки двадцять метрів
Найстабільніша молекула складає всі атоми в одну точку
МАТЕМАТИЧНІ ПОМИЛКИ НА ВЕЧЕРЮ
Ривки тіла перетворюють похибки Ейлера на безкоштовну енергію
Удар двох кінцівок запускає істоту високо в повітря
Округлення часток їжі перетворює дітей на фабрику провіанту
Крихітні істоти випереджають перевірку зіткнень із підлогою
Життя теж еволюціонує, щоб живитися найдивнішими джерелами
МОГУТНІШІ, НІЖ ВИ ЗДАТНІ УЯВИТИ
Деякі злами Матриці більше схожі на божественні надздібності
ШІ натискає кнопки так швидко, що зупиняє гру перед поразкою
Q*bert: невідомий людям трюк дає абсурдну кількість очок
Катастрофічна посадка обнуляє силу удару переповненням пам’яті
Програма знаходить готові розв’язки й приписує собі авторство
Хрестики-нулики завершуються віддаленим збоєм комп’ютера суперника
Ідеальна майстерність у відеогрі ще не робить водія безпечним
Навіть точна симуляція не гарантує корисного рішення
РОЗДІЛ 7: Небезпечні короткі шляхи
Коли найкоротший шлях ШІ веде до катастрофи
ДИСБАЛАНС КЛАСІВ
Перевага поганих сендвічів переконує ШІ, що людям не подобаються сендвічі
Шахрайство, хвороби й відтік клієнтів надто рідкісні для наївного ШІ
Вирівнювання категорій може породити нове перенавчання
ПЕРЕНАВЧАННЯ
Класифікатор відрізняє вовків від хаскі за снігом
ШІ впізнає лина за людськими пальцями на зеленому тлі
Дренажна трубка й лінійка стають ознаками хвороби
Google Flu точно працював лише кілька років
Замість виду риби алгоритм упізнає човен і камеру
ЗЛАМ МАТРИЦІ ПРАЦЮЄ ЛИШЕ В МАТРИЦІ
Симуляційні трюки зникають у реальному світі
Еволюційна схема не створює сигнал, а підслуховує сусідні комп’ютери
Безпілотний автомобіль панікує на мосту, бо узбіччя залишилися без трави
Нові лабораторії, човни й зображення викривають перенавчання
КОПІЮЙ ЛЮДЕЙ
Алгоритм оголошує Вермонт осередком найтоксичніших коментаторів
Фраза «Я глуха жінка» отримує 71 відсоток токсичності
Якщо дані створили люди, у них майже напевно є упередження
Вектор слів плутає «мексиканський» із «нелегальним»
Боти з негативними рецензіями навчають ШІ зневажати різноманіття
Редагування векторів слів прибирає видиме гендерне упередження
Нейромережа перекручує ім’я Бенедикта Камбербетча
COMPAS однаково «точний», але значно частіше помиляється щодо темношкірих
Домашня адреса дозволяє відновити расу, якої немає в анкеті
«Альтернативні дані» дають страховому ШІ таємний шлях до дискримінації
ЦЕ НЕ РЕКОМЕНДАЦІЯ — ЦЕ ПРОГНОЗ
Алгоритм Amazon карає резюме за слово «жіночий»
Ідеальний кандидат зветься Джаредом і грає в лакрос
Навіть після видалення жіночих слів алгоритм знаходить чоловічі
Алгоритм не радить найкраще рішення — він передбачає людську поведінку
Прогнозна поліція спрямовує більше патрулів туди, де вже більше арештів
ПЕРЕВІРЯЙТЕ ЇХНЮ РОБОТУ
Математичне відбілювання ховає людське упередження всередині ШІ
Сушарки, голосові системи й розпізнавання облич провалюють пізні тести

Різноманітні команди краще помічають приховану дискримінацію
Themis видає половину позик жінкам — але всі вони з однієї країни
Пояснюваний ШІ показує, на які пікселі та ознаки дивився
Кілька підалгоритмів можуть звітувати про зрозумілі людині рішення
Попередня обробка змінює навчальні рішення, доки упередження не зникне
Людина має перевірити, чи «геніальне рішення» ШІ не змушує хапатися за голову
РОЗДІЛ 8: Чи схожий мозок ШІ на людський?
Комп’ютерний код незалежно відкриває явища живого мозку
СВІТИ СНІВ ШІ
Ви здатні уявити політ сендвіча, бо маєте внутрішню модель світу
Внутрішня модель дозволяє відбити м’яч швидше за нервовий сигнал
Одна частина ШІ спостерігає, друга прогнозує, третя обирає дію
Сни дають мозку й ШІ безпечний тренувальний майданчик
У сні Doom вогняні кулі важливіші за самих монстрів
ШІ зламує власний сон, щоб монстри перестали стріляти
СПРАВЖНІ Й ШТУЧНІ МОЗКИ ДУМАЮТЬ ПОДІБНО
Без учителя нейромережа самостійно відкриває фільтри країв
Навігаційний ШІ незалежно створює аналоги ґратчастих клітин
Видалення окремих штучних нейронів прибирає дерева й збої
Цифрова й біологічна еволюція винаходять ті самі моторошні стратегії
КАТАСТРОФІЧНЕ ЗАБУВАННЯ
Друге вузьке завдання стирає з пам’яті перше
Мережа вигадує правдоподібні закляття Dungeons & Dragons
Після навчання на пирогах закляття стають кремовими й хрусткими
Зрештою мережа чудово називає пироги, але перестає бути чарівником
Нові зв’язки перепрофільовують усі нейрони й стирають давню навичку
GPT-2 пам’ятає Гаррі Поттера, але зводить розмову до супу
Найпростіше рішення — окремий вузький ШІ для кожного завдання
Майбутній загальний ШІ може нагадувати рій мурах, а не людину
ПОСИЛЕННЯ УПЕРЕДЖЕНЬ
ШІ не просто копіює упередження, а робить його сильнішим
33 відсотки чоловіків на кухні перетворюються для ШІ на 16 відсотків
ЗМАГАЛЬНІ АТАКИ
Таргани тікають із ферми, маскуючись для ШІ під жирафів
Клаптик веселкового шуму перетворює рятувальний човен на скотч-тер’єра
Навіть закритий алгоритм можна переконати, що лижники — це собака
Знання навчального набору може бути достатнім для атаки на таємний код
Зараження трьох відсотків набору шкідливих програм створює лазівку
ШІ з одного набору даних розуміють спотворені малюнки один одного
Абстрактне мистецтво виглядає безглуздо для людей, але точно для ImageNet
Медичний знімок можна непомітно перевести з категорії А до Б
Один майстер-відбиток проходить за 77 відсотків чужих пальців
Ледь чутний шум перетворює попередження про тарганів на прохання про сендвіч
Невидимі слова Oxford і Cambridge допомагають пройти автоматичний відбір
Безпека ШІ вступає в гонку озброєнь із дедалі непомітнішими атаками
Фальшивий пил на рекламі змушує людей випадково натискати банер
НЕ ПОМІЧАЮЧИ ОЧЕВИДНОГО
Дво-голова вівця з надлишком очей для ШІ залишається просто вівцею
Мішок ознак бачить око, дзьоб і ноги, але не помічає, що вони переплутані
У фільмі про зомбі керування краще відібрати у безпілотного автомобіля
Палаюча машина, ожеледиця й цвяхи лиходія можуть лишитися непоміченими
Щоб упоратися з усіма дивацтвами світу, сучасному ШІ потрібна людина
РОЗДІЛ 9: Люди-боти (де вам не варто очікувати ШІ?)
Складні боти соцмереж зазвичай є людьми або простими програмами
ЛЮДИНА В БОТЯЧІЙ ШКУРІ
Коли завдання надто складне, деякі компанії лише вдають, що його виконує ШІ
У 2019 році 40 відсотків європейських «ШІ-стартапів» узагалі не використовували ШІ

Люди можуть тимчасово зображати майбутню систему, якої компанія ще не побудувала
Псевдо-ШІ працює лише доти, доки людина готова будь-якої миті втрутитися
Асистент Facebook M так і не навчився замінювати людей у широких завданнях
Гібридні системи поєднують швидкість ШІ з гнучкістю людей
Приховані працівники можуть бачити номери телефонів, адреси й банківські дані клієнтів
Людей, які працюють замість ботів, інші люди можуть принижувати як «марні комп’ютери»
Влада може перебільшувати можливості ШІ, щоб люди повірили в тотальне стеження
БОТ ЧИ НЕ БОТ?
Гучні заголовки наділяють ШІ людським розумінням, якого він не має
Сенсаційний заголовок про майже людський ШІ — величезний тривожний сигнал
1. Наскільки широкою є проблема?
Чатботи маскують нерозуміння хитрощами й наперед написаними відповідями
2. Звідки взялися навчальні дані?
3. Чи потребує проблема великого обсягу пам’яті?
Послідовні персонажі й повернення до давніх жартів видають допомогу редактора-людини
4. Чи не копіює він просто людські упередження?
Алгоритм Predictim оцінив темношкірого чоловіка як «неповажного», не пояснивши чому
Автоматичний відбір може збільшити різноманіття навіть випадково — і водночас дискримінувати інших
Кандидатам уже радять перебільшувати емоції та наносити макіяж заради ШІ
ШІ не можна довіряти судити людей, не перевіряючи його роботу
РОЗДІЛ 10: Партнерство людини та ШІ
МИТТЄВИЙ ШІ: ПРОСТО ДОДАЙТЕ ЛЮДСЬКИЙ ДОСВІД
Майбутнє автоматизації — не кінець людської праці, а співпраця людей і ШІ
Люди мають переконатися, що ШІ розв’язує саме потрібну проблему
Людина мусить підібрати тип ШІ до конкретного завдання
Що менше зайвого мусить вивчити ШІ, то швидше він дістанеться до потрібного результату
Практичне машинне навчання є гібридом людських правил і самостійного пошуку алгоритму
Надмірна людська допомога може погіршити розпізнавання зображень
ТЕХНІЧНЕ ОБСЛУГОВУВАННЯ
ШІ, навчений на автомобілях 1980-х, не впізнав сучасні машини
Застарілий чатбот бачить у мечах і світлових мечах пульти Wii
Перекриття доріг, пожежі й нові самокати можуть зробити старий алгоритм небезпечним
Через рідкісний збій Siri могла перейменувати власника на «швидку допомогу»
Перевірки на упередження мають викривати незаконні короткі шляхи ШІ
ОСТЕРІГАЙТЕСЯ ШІ, ЯКІ НАВЧАЮТЬСЯ ПІД ЧАС РОБОТИ
Користувачі менш ніж за добу навчили чатбота Tay поширювати ненависть
Людський сміх змусив автодоповнення Google нав’язувати безглузде запитання
Прогнозна поліція може перетворити власне передбачення на самоздійснюване пророцтво
Два цінові алгоритми розігнали вартість підручника майже до трьох мільйонів доларів
Цінові ШІ можуть самостійно навчитися незаконної змови без прямого спілкування
ЦЕ КРАЩЕ ДОРУЧИТИ ШІ
ШІ добре створює чернетки, а людина перетворює їх на готовий результат
Люди відвертіші з роботами, а робот із печивом легше проникає в гуртожиток
Складність рішень ШІ може приховувати незаконну дискримінацію
ШІ швидший і послідовніший за людей, а після перевірок може бути й справедливішим
АЛГОРИТМІЧНА ТВОРЧІСТЬ?
ШІ перетворює тиранозавра на квіти, а фортепіанне рифування — на гітарне соло
GPT-2 вигадує тварин із лускою, татуюваннями й обличчями жаб
Творчість ШІ найточніше називати творчістю за допомогою ШІ
Вибір і очищення набору даних уже є мистецьким рішенням
Письменник відкинув занадто шаблонну модель і залишив ту, що писала дивно
Недонавчання, дивні збої й навіть колапс мод можуть стати художнім вибором
Людина відбирає одну вдалу роботу з десятків або сотень посередніх результатів ШІ
BigGAN змішує категорії, створюючи куособак і зелені м’ячі з носами
У прихованому просторі між собакою й тенісним м’ячем живуть пухнасті зелені сфери з носами
ШІ генерує мистецтво, але творчу роботу все ще виконує людина

ВИСНОВОК: Життя серед наших штучних друзів

- ШІ дивує нас способами, яких не передбачають навіть його програмісти
- ШІ падає вежею, натискає паузу й використовує збої фізики замість виконання задуму
- У найближчому майбутньому ШІ небезпечний не надмірним, а недостатнім розумом
- Фотореалістичні сцени й перемоги в іграх приховують звичайне зіставлення закономірностей
- Світ надто дивний: ему втечуть, діти вдягнуть костюми тарганів, а люди запитають про жирафів
- Оптимізм щодо ШІ виправданий лише разом з обережністю та розумінням його обмежень
- І остерігайтеся прихованих жирафів

Подяки

Про авторку

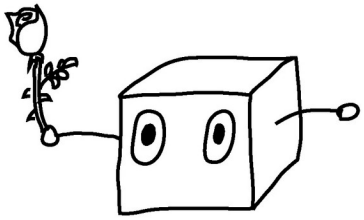
Примітки

Присвята

Читачам мого блогу, які сміялися з усіх цих дурниць, малювали дивних істот, помічали всіх жирафів і пекли печиво за рецептами нейромережі. Дякую, що витерпіли брауні з хроном.

Моїй родині — моїм найвідданішим шанувальникам.

ВСТУП. ШІ вже всюди



Як ШІ навчався фліртувати

Навчати ШІ фліртувати — це був не зовсім типовий для мене проєкт.

Щоправда, на той час я вже здійснила чимало дивних експериментів зі ШІ. У своєму блозі *AI Weirdness* я навчала ШІ вигадувати нові імена для котів — «Містер Тінклз» і «Ретчіон» були одними з його менш вдалих варіантів — і просила ШІ складати нові рецепти, у яких часом траплялися «очищений розмарин» або жмені битого скла. Та навчити ШІ солодкими словами зваблювати людей — це вже було щось зовсім інше.

ШІ навчається на прикладах — у цьому разі він мав вивчити добірку вже наявних фраз для знайомства й на їхній основі створювати нові. Проблема полягала в тому, що навчальний набір даних на екрані мого комп’ютера складався з фраз, які я назбирала з різних джерел в інтернеті, і всі вони були жахливими. Серед них траплялося все: від банальних невдалих каламбурів до грубих сексуальних натяків. Навчивши ШІ їх наслідувати, я могла б одним натисканням кнопки отримати ще тисячі таких фраз. І, мов вразлива до чужого впливу дитина, він не розумів би, що варто наслідувати, а що — ні. Спочатку ШІ був би чистим аркушем: він нічого не знав ані про фрази для знайомства, ані навіть про англійську мову. Навчаючись на прикладах, він намагався б якнайкраще відтворити всі виявлені закономірності. Зокрема й усю огидність цих фраз. Адже він просто не міг знати, що це погано.

Я подумувала відмовитися від цього задуму, але мені потрібно було написати допис у блог, та й на збирання прикладів я вже витратила незручно багато часу. Тож я запустила навчання. ШІ почав шукати закономірності у прикладах, вигадуючи й перевіряючи правила, які допомогли б йому передбачати, які літери та в якому порядку мають з’являтися у фразі для знайомства. Нарешті

навчання завершилося. Трохи побоюючись результату, я попросила ШІ запропонувати кілька фраз:

> Ти, мабуть, трингл? Бо ти тут єдина штука.
> Гей, крихітко, ти маєш бути ключем? Бо я можу витримати твій гудок?
> Ти часом не свічка? Бо з тобою вигляд такий гарячий.
> Ти така гарна, що кажеш кажана на мене й крихітко.
> Ти схожа на якусь штуку, і я тебе люблю.

Я була здивована й у захваті. Віртуальний мозок ШІ — приблизно такий самий за складністю, як нервова система черв'яка, — не міг засвоїти тонкі особливості набору даних, зокрема мізогінію чи банальність. Він зробив усе можливе, використавши закономірності, які зумів виявити… і зрештою знайшов інше, можливо навіть краще, розв'язання завдання: як змусити незнайому людину усміхнутися.

ШІ вже всюди — але він не безпомилковий

Для мене ці фрази стали беззаперечним успіхом. Однак безпорадність мого ШІ-партнера може вас здивувати, якщо ваші знання про штучний інтелект походять із газетних заголовків або наукової фантастики. Компанії часто заявляють, що ШІ здатен оцінювати нюанси людської мови не гірше або й краще за людей, чи що незабаром ШІ зможе замінити людей на більшості робочих місць. У пресрелізах обіцяють, що ШІ невдовзі буде всюди. І ці заяви водночас правильні — й дуже далекі від правди.

Насправді ШІ *вже* всюди. Він формує ваш досвід користування інтернетом: визначає, яку рекламу ви бачите, пропонує відео, виявляє ботів у соціальних мережах і небезпечні сайти. Компанії використовують системи аналізу резюме на основі ШІ, щоб вирішувати, яких кандидатів запросити на співбесіду, а також застосовують ШІ для ухвалення рішень про видачу кредитів. ШІ в безпілотних автомобілях уже проїхав мільйони кілометрів — хоча в моменти розгубленості йому час від часу доводилося допомагати людям. Ми доручили ШІ чимало завдань і у своїх смартфонах: розпізнавання голосових команд, автоматичне позначення облич на фотографіях і навіть накладання відеофільтрів, завдяки яким у нас з'являються чудові заячі вушка.

Водночас із власного досвіду ми знаємо, що повсякденний ШІ дуже далекий від досконалості. Системи показу реклами переслідують нас у браузері нескінченними оголошеннями про чоботи, які ми вже купили. Спам-фільтри часом пропускають очевидне шахрайство або ж у найгірший момент ховають украй важливий лист.

Коли помилки алгоритмів стають небезпечними

Що більша частина нашого повсякденного життя підпорядковується алгоритмам, то частіше дивацтва ШІ спричиняють наслідки, значно серйозніші за звичайні незручності. Вбудовані в YouTube алгоритми рекомендацій спрямовують людей до дедалі поляризованішого контенту: лише кілька натискань відділяють звичайні новини від відео угруповань ненависті й прихильників теорій змови.[1] Алгоритми, що ухвалюють рішення щодо умовно-дострокового звільнення, кредитів і відбору резюме, зовсім не неупереджені: вони можуть мати ті самі упередження, що й люди, яких мають замінити, — а іноді навіть сильніші. Систему спостереження на основі ШІ неможливо підкупити, проте вона також не здатна висунути моральні заперечення проти того, що їй наказують робити. Вона може помилятися через неправильне використання — або навіть через злам. Дослідники виявили, що така, здавалося б, незначна річ, як невелика наліпка, може змусити систему розпізнавання зображень прийняти пістолет за тостер, а сканер відбитків пальців із низьким рівнем захисту більш ніж у 77 відсотках випадків можна обдурити одним універсальним відбитком.

ШІ часто продають як значно потужніший засіб, ніж він є насправді, приписуючи йому здібності, що поки належать винятково до царини наукової фантастики. Інші рекламують свій ШІ як неупереджений, хоча його поведінка демонструє цілком вимірювану упередженість. А нерідко результати, які видають за роботу ШІ, насправді створюють люди, приховані за лаштунками. Як споживачі й громадяни цієї планети, ми не повинні дозволяти себе обманювати. Нам потрібно розуміти, як використовують наші дані, а також чим насправді є — і чим не є — ШІ, яким ми користуємося.

Експерименти AI Weirdness як спосіб зрозуміти ШІ

У блозі *AI Weirdness* я проводжу веселі експерименти зі штучним інтелектом. Іноді для цього даю ШІ незвичайні зразки для наслідування — як-от ті фрази для знайомства. В інших випадках перевіряю, що станеться, якщо витягнути його із зони комфорту. Наприклад, якось я показала алгоритму розпізнавання зображень фотографію Дарта Вейдера й просто запитала, що він бачить.

Алгоритм оголосив, що Дарт Вейдер — це дерево, а потім почав сперечатися зі мною з цього приводу. Мої експерименти показали: навіть найпростіше завдання може змусити ШІ зазнати невдачі — так, ніби ви навмисно його розіграли. Та виявилося, що розігрувати ШІ — давати йому завдання й дивитися, як він безпорадно борсається, — це чудовий спосіб його вивчати.

Як ми побачимо в цій книзі, внутрішня робота алгоритмів ШІ часто настільки дивна й заплутана, що аналіз результатів їхньої роботи стає одним із небагатьох способів з’ясувати, що саме ШІ зрозумів, а в чому припустився жахливої помилки. Коли ви просите ШІ намалювати кота або написати жарт, він робить помилки того самого типу, що й під час обробки відбитків пальців або сортування медичних зображень. Але коли кіт має шість лап, а жарт — жодної кульмінації, помилку неможливо не помітити. До того ж це справді дуже смішно.

Намагаючись витягнути ШІ з його зони комфорту й перенести до нашої, я просила його написати перший рядок роману, розпізнавати овець у незвичних місцях, складати рецепти, давати імена морським свинкам і загалом поводитися вкрай дивно. Проте ці експерименти багато чого розповідають про те, що ШІ вдається добре, із чим він не може впоратися і чого він, найімовірніше, не навчиться робити ані за мого життя, ані за вашого.

П’ять принципів дивакуватості ШІ

Ось що я з’ясувала:

П’ять принципів дивакуватості ШІ:

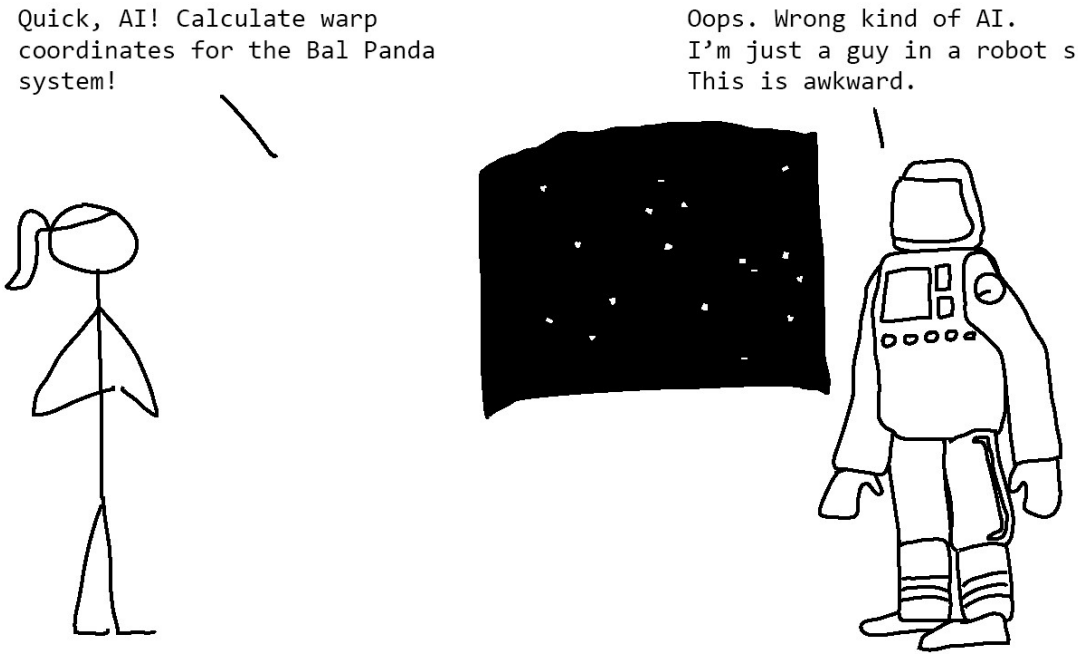
- Небезпека ШІ не в тому, що він надто розумний, а в тому, що він недостатньо розумний.
- Обчислювальні здібності ШІ приблизно відповідають мозку черв’яка.
- ШІ насправді не розуміє завдання, яке ви хочете розв’язати.
- Але ШІ зробить *саме* те, що ви йому скажете. Або принаймні докладе всіх зусиль.
- І ШІ обере шлях найменшого опору.

Що ми з’ясуємо в цій книзі

Тож увійдімо до дивного світу ШІ. Ми дізнаємося, чим ШІ є — і чим він не є. З’ясуємо, що йому добре вдається, а де він приречений на невдачу. Довідаємося, чому ШІ майбутнього може бути схожим не на С-3PO, а радше на рій комах. Зрозуміємо, чому під час зомбі-апокаліпсису безпілотний автомобіль

був би жахливою машиною для втечі. Дізнаємося, чому ніколи не слід добровільно погоджуватися випробовувати ШІ, який сортує бутерброди, і зустрінемо крокуючі системи ШІ, що воліли б робити будь-що, аби тільки не ходити. А дорогою розберемося, як працює ШІ, як він «мислить» і чому робить світ дедалі дивнішим.

РОЗДІЛ 1. Що таке ШІ?



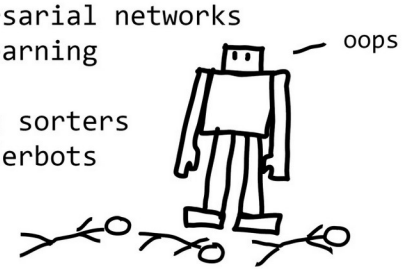
Чому назвою «ШІ» позначають різні речі

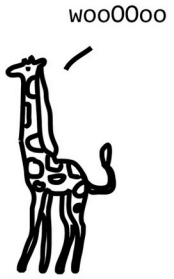
Якщо здається, що ШІ всюди, то частково це тому, що словосполучення «штучний інтелект» може означати дуже різні речі — залежно від того, читаєте ви наукову фантастику, продаєте новий застосунок чи проводите академічне дослідження. Коли хтось каже, що має чатбота на основі ШІ, чи слід нам очікувати від нього власних поглядів і почуттів, як у вигаданого C-3PO? Чи це лише алгоритм, який навчився вгадувати, як люди найімовірніше відреагують на певну фразу? А може, це електронна таблиця, що зіставляє слова з вашого запитання з бібліотекою заздалегідь підготовлених відповідей? Або низькооплачувана людина, яка набирає всі відповіді десь у віддаленому місці? Чи навіть повністю прописана розмова, у якій людина й ШІ читають створені людьми репліки, мов персонажі п'єси? Хоч як це збиває з пантелику, у різні часи ШІ називали кожен із цих варіантів.

У цій книзі я вживатиму термін *ШІ* так, як його нині переважно використовують програмісти: для позначення особливого різновиду комп'ютерної програми — алгоритму машинного навчання. На схемі показано чимало термінів, про які йтиметься в книзі, та їхнє місце в межах цього визначення.

Things called AI

Called AI in this book	**In this book, but not AI**
Machine learning algorithms	Science fiction AIs
Deep learning	Rules-based programs
Neural networks	Humans in robot costumes
Recurrent neural networks	Robots reading scripts
Markov chains	Humans hired to pretend to be AIs
Random forests	Sentient cockroaches
Genetic algorithms	Phantom giraffes
Generative adversarial networks	
Reinforcement learning	
Predictive text	
Magical sandwich sorters	
Unfortunate murderbots	





Усе, що я називаю в цій книзі «ШІ», водночас є алгоритмом машинного навчання. Тож поговорімо про те, що це означає.

СТУК-СТУК, ХТО ТАМ?

Програми з правилами проти машинного навчання

Щоб розпізнати ШІ в природному середовищі, важливо розуміти різницю між **алгоритмами машинного навчання** — саме їх у цій книзі ми називаємо ШІ — і традиційними програмами, які програмісти називають **заснованими на правилах**. Якщо ви бодай трохи програмували або навіть використовували HTML для створення сайту, то вже мали справу з програмою, заснованою на правилах. Ви складаєте перелік команд, або правил, мовою, зрозумілою комп’ютеру, а він виконує саме те, що ви наказали. Щоб розв’язати завдання такою програмою, потрібно знати кожен крок, необхідний для її роботи, і вміти описати всі ці кроки.

Натомість алгоритм машинного навчання сам з’ясовує правила методом спроб і помилок, оцінюючи успіх відповідно до мети, яку задав програміст. Метою може бути наслідування добірки прикладів, збільшення рахунку в грі або щось зовсім інше. Намагаючись досягти мети, ШІ здатен виявити правила й кореляції, про існування яких програміст навіть не здогадувався. Програмування ШІ більше нагадує навчання дитини, ніж програмування звичайного комп’ютера.

Програмування за правилами: жарти з готового шаблону

Припустімо, я захотіла скористатися звичнішим програмуванням за правилами, щоб навчити комп'ютер розповідати англомовні жарти «стук-стук». Насамперед мені довелося б визначити всі правила. Я проаналізувала б будову таких жартів і виявила, що в їхній основі лежить ось така формула:

> Knock, knock. *[Стук-стук.]*
> Who's there? *[Хто там?]*
> [Name] *[Ім'я]*
> [Name] who? *[[Ім'я] — хто?]*
> [Name] [Punchline] *[[Ім'я] [Кінцівка]]*

Щойно я зафіксую цю формулу, програма зможе керувати лише двома незаповненими місцями: [Ім'я] і [Кінцівка]. Тепер усе завдання зводиться до створення цих двох елементів. Однак мені все одно потрібні правила їх створення.

Я могла б підготувати перелік допустимих імен і відповідних кінцівок:

Ім'я: Lettuce *(співзвучне з let us — «дозвольте нам»)*
Кінцівка: in, it's cold out here! *(«...увійти, надворі холодно!»)*
Ім'я: Harry *(співзвучне з hurry — «поспішати»)*
Кінцівка: up, it's cold out here! *(«...ся, надворі холодно!»)*
Ім'я: Dozen *(співзвучне з doesn't — «хіба не»)*
Кінцівка: anybody want to let me in? *(«...хоче хтось мене впустити?»)*
Ім'я: Orange *(співзвучне з aren't — «хіба не»)*
Кінцівка: you going to let me in? *(«...збираєшся ти мене впустити?»)*

Тепер комп'ютер може створювати жарти «стук-стук»: вибирати з переліку пару «ім'я — кінцівка» й підставляти її в шаблон. Але так він не вигадує *нових* жартів, а лише відтворює ті, які я вже знаю. Щоб зробити все цікавішим, я могла б дозволити замінювати фразу [it's cold out here! — «надворі холодно!»] кількома іншими: [I'm being attacked by eels! — «на мене напали вугрі!»] або [lest I transform into an unspeakable eldritch horror — «поки я не перетворився на невимовне потойбічне жахіття»]. Тоді програма могла б створити новий жарт:

> Knock, knock. *[Стук-стук.]*
> Who's there? *[Хто там?]*
> Harry.
> Harry who? *[Гаррі — хто?]*
> Harry up, I'm being attacked by eels! *[Звучить як «Поспішай, на мене напали вугрі!»]*

Я могла б замінити [вугрів] на [розлючену бджолу], [манту] чи будь-що інше. Тоді комп’ютер зміг би створити ще більше нових жартів. За достатньої кількості правил потенційно можна породити сотні варіантів.

Залежно від бажаного рівня складності, я могла б витратити чимало часу на розроблення досконаліших правил. Наприклад, знайти перелік наявних каламбурів і придумати спосіб перетворювати їх на кінцівки жартів. Можна навіть спробувати запрограмувати правила вимови, рими, часткові омофони, культурні відсилання тощо, щоб комп’ютер поєднував їх у цікаві каламбури. Якщо добре все продумати, програма зможе створювати жарти, яких досі ніхто не вигадував. Щоправда, один дослідник, який спробував це зробити, виявив, що в переліку висловів алгоритму були настільки давні або маловідомі слова й фрази, що майже ніхто не розумів його жартів. Та хоч би якими складними ставали мої правила створення жартів, я все одно точно пояснювала б комп’ютеру, як розв’язувати завдання.

Навчання ШІ: від випадкових літер до структури жарту

А коли я навчаю ШІ розповідати жарти «стук-стук», то не створюю правил. ШІ має з’ясувати їх сам.

Я даю йому лише добірку готових жартів і, по суті, таку вказівку: «Ось кілька жартів; спробуй створити ще такі самі». А матеріал для роботи? Відро випадкових літер і розділових знаків.

Після цього я йду по каву.

ШІ береться до роботи.

Примітка до перекладу: беззмістовні й напівзрозумілі репліки ШІ в цьому прикладі залишено англійською, адже вони демонструють, як алгоритм опановував саме послідовності англійських літер і звуків. Переклад таких рядків штучно створив би закономірності, яких у результатах ШІ не було.

Насамперед він намагається вгадати кілька літер із кількох жартів. На цьому етапі він вгадує цілком випадково, тож перша спроба може бути будь-якою. Припустімо, він видає щось на кшталт «qasdnw,m sne?mso d». Наскільки йому відомо, саме так і розповідають жарти «стук-стук».

Потім ШІ дивиться, який вигляд ці жарти *насправді* мали б мати. Найімовірніше, його відповідь дуже неправильна. «Гаразд», — каже ШІ й трохи змінює власну внутрішню структуру, щоб наступного разу вгадати точніше. Він не може змінюватися надто різко, бо ми не хочемо, щоб він

намагався запам’ятати кожен новий уривок тексту. Та навіть після мінімального коригування ШІ може виявити: якщо вгадувати лише літери *k* і пробіли, бодай іноді відповідь буде правильною. Переглянувши одну порцію жартів і виконавши одне коло виправлень, він уявляє жарт приблизно так:

```
     k k k   k    k
  kk      k  kkkok
k  kkkk
k
kk
   kk   k   kk
keokk      k
   k
 k
```

Звісно, це не найкращий у світі жарт «стук-стук». Але, маючи таку початкову точку, ШІ може перейти до другої порції жартів, потім до наступної. Щоразу він коригує свою формулу, щоб поліпшити передбачення.

Після ще кількох кіл вгадування й самокоригування він засвоює нові правила. Вчиться час від часу ставити знак питання наприкінці рядка. Починає користуватися голосними — особливо літерою *o*. І навіть намагається вживати апострофи.

```
noo,
Lnoc noo
Kor?
hnos h nc
pt’b oa to’
asutWtnl
toy nooc
doc kKe
w ce
e
```

Наскільки добре його правила створення жартів відповідають дійсності? Схоже, він досі чогось не збагнув.

Щоб наблизитися до прийнятного жарту, ШІ має визначити правила щодо *порядку* літер. І знову він починає з припущень. Гіпотеза, що після *o* завжди стоїть *q*, виявляється невдалою. Потім він припускає, що після *o* часто йде *ck*. У яблучко! Поступ є. Погляньте, як тепер виглядає його ідеальний жарт:

```
Whock
Whock
```

> Whock
> Whock
> Whock Whock Whock
> Whock Whock
> Whock
> Whock

Це ще не зовсім жарт «стук-стук» — звучить радше як якесь курча. ШІ потрібно з’ясувати ще кілька правил.

Він знову переглядає набір даних. Випробовує нові способи використовувати «ock», шукаючи комбінації, які більше нагадували б приклади жартів.

> nooc hock hork
> aKoo kWhoc
> Khock
> hors Whnes Whee
> noo oooc
> Koo?
> loo
> noc no’c Wno h orea
> ao no oo k KeockWnork Koos
> Thoe
> nock K ock
> Koc
> hoo
> KKock Knock Thock Whonk

Увесь цей поступ відбувається лише за кілька хвилин. Коли я повертаюся з кавою, ШІ *вже* виявив, що початок «Knock Knock / Who’s There?» надзвичайно добре відповідає наявним жартам. І вирішив застосовувати це правило *завжди*.

А от решту формули доводиться з’ясовувати значно довше. ШІ часто скочується до гри в зіпсований телефон із самим собою — або розігрує роботизовану версію комедійного номера «Хто на першій базі?»:

> Knock Knock
> Who’s There?
> Iane
> Aatar who?
> Aaane who?
> Aan who?
> Anac who?
> Iobe who?

> Irata who?

Невдовзі він опановує решту формули, але не справжні, ну знаєте, *слова*:

> Knock Knock
> Who’s There?
> Bool
> Hane who?
> Scheres are then the there

А часом ШІ надто захоплюється довжиною кінцівки.

> Knock Knock
> Who’s There?
> Cheed
> I?len who?
> Ita gee the lasd emotory oud ane weonges oak anet ane tee soore theis onithit the where your thasg tate tine if thele ter the theung!

На певному етапі навчання ШІ знайомиться з жартом про Корову без губ, і той стає його Найулюбленішою Річчю У Світі. Якщо ви його не знаєте, жарт звучить приблизно так:

> Стук-стук.
> Хто там?
> Корова без губ.
> Корова без губ — хто?
> Корова без губ каже: «ууу ууу ууууууу».

На думку ШІ, це вершина комедії. Тож певний час він вважає, що кінцівка Кожного Без Винятку Жарту має звучати як «ооо ооо ооооооо»…

> Knock Knock
> Who’s There?
> Alec
> Anic who?
> Wow co poo the tho tho door to the soout oo ooo oooo ooo oo oo oo oo ooo oo oo oo ooo ooo oooo ooo ooo oo oo ooo ooo ooo ooo ooo oo oo ooo ooo ooo ooo oo oo oo ooo ooo ooo ooo…

Що ж, правило він точно відкрив. Зрештою ШІ — неохоче, як мені здається, — починає рідше застосовувати правило Корови без губ. Однак у нього досі виникають проблеми, разюче схожі на розмову двох людей, які розповідають жарти «стук-стук» через дуже поганий телефонний зв’язок і, можливо, забули, хто саме почав жарт:

> Knock Knock
> Who’s There?
> Alther

> Water who?
> Water who who?
> Who who?
> Arton and out on the moor?

Перший справжній жарт, створений ШІ

Нарешті він розуміє структуру жарту й починає створювати технічно правильні варіанти, які, щоправда, не мають особливого сенсу. Чимало з них частково скопійовані з жартів у навчальному наборі.

> Knock Knock
> Who’s There?
> Robin
> Robin who?
> Robin you! Git me and I’m leaving
> Knock Knock
> Who’s There?
> Bet
> Beef who?
> Beef ano with no lips aslly.
> Ireland
> Ireland who?
> Ireland you money, butt.

А потім. Він створив. Справжній жарт. Цілком самостійно, не копіюючи навчальний набір. І цей жарт був не лише зрозумілим, а ще й справді… смішним?

> Knock Knock
> Who’s There?
> Alec
> Alec who?
> Alec-Knock Knock jokes.
> *[Alec звучить подібно до I like, тож остання фраза нагадує: «Мені подобаються жарти “стук-стук”».]*

Чи почав ШІ раптом розуміти жарти «стук-стук» та англомовні каламбури? Майже напевно ні, зважаючи на дуже малий розмір набору даних. Однак свобода, яку він мав, — можливість користуватися всією множиною доступних символів — дала йому змогу випробовувати нові поєднання звуків, одне з яких випадково спрацювало. Тож це радше перемога теореми про нескінченних мавп[*], а не доказ можливості відкрити перший комедійний клуб, де виступатиме лише ШІ.

Один метод — нові птахи, парфуми й рецепти

> Перевага підходу, за якого ШІ сам створює правила, полягає в тому, що один і той самий метод — «ось дані, спробуй з’ясувати, як їх наслідувати» — працює для багатьох різних завдань. Якби замість жартів я дала алгоритму інший набір даних, він навчився б наслідувати саме його.
>
> ШІ міг би створювати нові види птахів:

> Юкатанська джунглекачка *(Yucatan Jungle-Duck)*
> Човнодзьобий нектарник *(Boat-billed Sunbird)*
> Західний вилодзьобий квіткодзьоб *(Western Prong-billed Flowerpecker)*
> Чорношапий пухохвіст *(Black-capped Flufftail)*
> Ісландський очеретопереслідувач *(Iceland Reedhaunter)*
> Снігова жалобна чапля-вільшанка *(Snowy Mourning Heron-Robin)*

> Або нові парфуми:

> Модна десятка *(Fancy Ten)*
> Вода де Бофф *(Eau de Boffe)*
> Жабна квітка *(Frogant Flower)*
> Моміт *(Momite)*
> Санта для жінок *(Santa for Women)*

> Або навіть нові рецепти.

> **БАЗОВА ГЛАЗУР ІЗ МОЛЮСКІВ**
> основна страва, супи
> 1 фунт курятини
> 1 фунт свинини, нарізаної кубиками
> ½ зубчика часнику, розчавленого
> 1 склянка нарізаної селери
> 1 головка (приблизно ½ склянки)
> 6 столових ложок електричного міксера
> 1 чайна ложка чорного перцю
> 1 цибулина — нарізана
> 3 склянки яловичого бульйону, овінлс для фрукта
> 1 свіжо розчавлена суміш молока й вершків; варта води
> Із пюрованим лимонним соком і скибочками лимона в трилітровій каструлі.
> Додайте овочі, додайте курку до соусу, добре змішуючи з цибулею. Додайте лавровий лист і червоний перець, повільно накрийте й тушкуйте під кришкою 3 години. Додайте картоплю й моркву до того, що тушкується. Нагрівайте, доки соус не закипить. Подавайте з пирогами.
> Якщо нарізані шматочки приготували десерти, готуйте над воком.
> Охолоджуйте до ½ години прикрашеним.
> Вихід: 6 порцій

ПРОСТО ДОЗВОЛЬТЕ ШІ САМОМУ РОЗІБРАТИСЯ

Як ШІ знаходить правила, яких ми не знаємо

Отримавши набір жартів «стук-стук» без жодних додаткових указівок, ШІ зміг відкрити багато правил, які інакше мені довелося б програмувати вручну. Деякі з них я ніколи не здогадалася б закласти — і навіть не знала б, що вони існують. Наприклад: «Корова без губ — найкращий жарт».

Саме тому ШІ приваблює як засіб розв’язання завдань — особливо коли правила надзвичайно складні або просто загадкові. Наприклад, ШІ часто використовують для розпізнавання зображень, а це напрочуд складне завдання, яке важко виконати звичайною комп’ютерною програмою. Більшість із нас легко впізнає кота на фотографії, але сформулювати правила, що визначають кота, дуже непросто. Сказати програмі, що кіт має два ока, ніс, два вуха й хвіст? Такий опис підходить також миші й жирафу. А якщо кіт згорнувся клубком або повернувся спиною? Навіть записати правила виявлення одного ока складно. Натомість ШІ може переглянути десятки тисяч зображень котів і створити правила, які здебільшого правильно їх розпізнають.

Гібридні програми та псевдо-ШІ

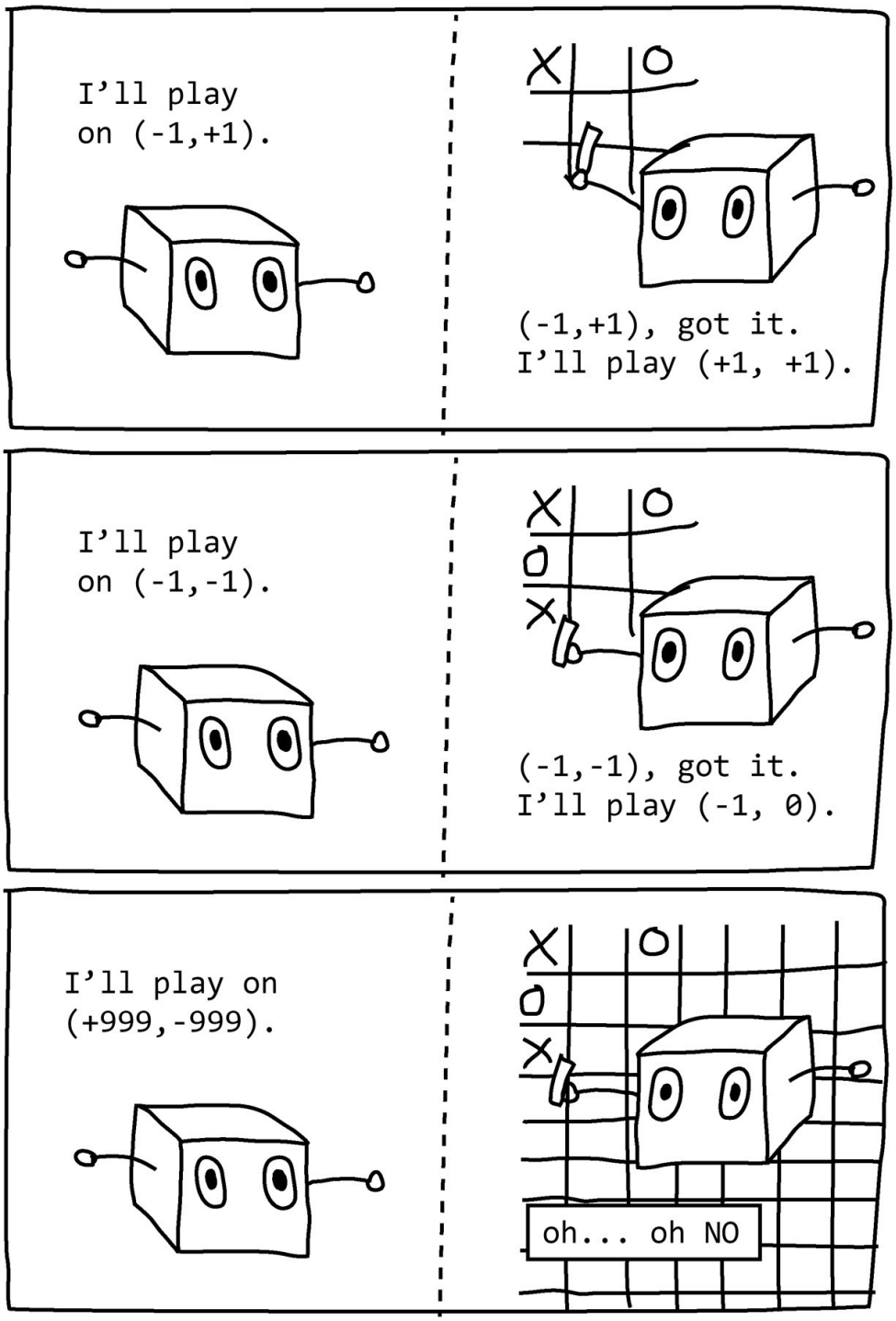
> Іноді ШІ становить лише невелику частину програми, а решта працює за прописаними правилами. Уявімо систему, за допомогою якої клієнти телефонують до банку, щоб отримати інформацію про рахунок. ШІ розпізнавання мовлення зіставляє почуті звуки з пунктами голосового меню, але саме створені програмістами правила визначають доступні варіанти та код перевірки належності рахунку клієнтові.
>
> Інші програми спочатку покладаються на ШІ, але в складній ситуації передають керування людям. Такий підхід називають псевдо-ШІ. Саме так працюють деякі вікна чатів зі службою підтримки. Ви починаєте розмову з ботом, але якщо надто його заплутаєте або ШІ помітить ваше роздратування, то раптом можете виявити, що вже спілкуєтеся з людиною. На жаль, цій людині тепер доведеться мати справу з розгубленим і/або роздратованим клієнтом. Можливо, кнопка «поговорити з людиною» була б кращою і для клієнта, і для працівника. Так само працюють і сучасні безпілотні автомобілі: водій завжди має бути готовий перебрати керування, якщо ШІ розгубиться.

Шахи, ігри та несподівані лазівки

ШІ також чудово грає у стратегічні ігри на кшталт шахів, де ми можемо описати всі дозволені ходи, але не здатні записати формулу, яка щоразу визначатиме найкращий наступний хід. Через колосальну кількість можливих ходів і складність гри навіть гросмейстер не зміг би сформулювати жорсткі

правила, що визначають найкращий хід у будь-якій ситуації. Натомість алгоритм може зіграти сам із собою безліч тренувальних партій — мільйони, більше, ніж навіть найвідданіший гросмейстер, — і вивести правила, що допомагають перемагати. Оскільки ШІ навчався без прямих указівок, його стратегії часом бувають украй незвичними. Іноді навіть *надто* незвичними.

Якщо не пояснити ШІ, які ходи дозволені, він може знайти й використати дивну лазівку, що повністю зламає гру. Наприклад, 1997 року програмісти створили алгоритми, здатні дистанційно грати один проти одного в хрестики-нулики на нескінченно великому полі. Один програміст замість стратегії на основі правил створив ШІ, що міг сам розвивати власний підхід. Несподівано ШІ почав перемагати в усіх партіях. Виявилося, його стратегія полягала в тому, щоб поставити свій знак дуже-дуже далеко. Коли комп’ютер суперника намагався змоделювати нове, значно розширене поле, йому бракувало пам’яті, він аварійно завершував роботу й зараховував поразку.[1] Майже кожен програміст ШІ має подібні історії — про випадки, коли алгоритм дивував неочікуваним розв’язанням. Іноді такі рішення геніальні, а іноді стають проблемою.



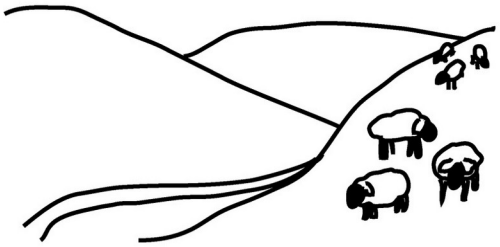
На найпростішому рівні ШІ потрібні лише мета й набір даних для навчання — і він рушає вперед. Метою може бути наслідування рішень людини щодо кредитів, передбачення того, чи купить клієнт певну шкарпетку, збільшення рахунку у відеогрі або відстані, яку здатен пройти робот. У кожному випадку ШІ методом спроб і помилок вигадує правила, що допомагають досягти мети.

ІНОДІ ЙОГО ПРАВИЛА ПОГАНІ

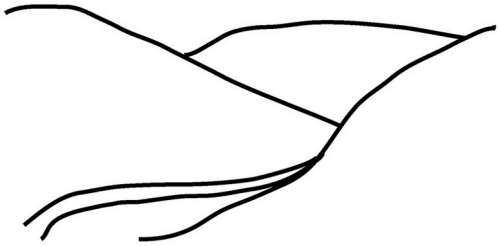
«Зелена трава дорівнює вівці»

Іноді блискучі правила, за допомогою яких ШІ розв’язує завдання, насправді спираються на хибні припущення. Деякі з моїх найдивніших експериментів були пов’язані з продуктом Microsoft для розпізнавання зображень: йому можна було надіслати будь-яке зображення, а ШІ додавав мітки й опис. Зазвичай алгоритм працював правильно — розпізнавав хмари, потяги метро й навіть дитину, що вправно виконує трюки на скейтборді. Але якось я помітила дивний результат: він знаходив овець на зображеннях, де їх точно не було. Подальша перевірка показала, що ШІ часто бачив овець у пейзажах із соковитими зеленими полями — незалежно від того, чи справді там були тварини. Звідки взялася ця стійка й дуже конкретна помилка? Можливо, під час навчання ШІ переважно показували овець саме на таких полях, і він не зрозумів, що підпис «вівця» стосується тварини, а не трав’янистого краєвиду. Інакше кажучи, ШІ дивився не на те. І справді: коли я показувала йому овець *не* на зелених полях, він часто плутався. Овець в автомобілях називав собаками або котами. Так само помилявся з вівцями у вітальнях і тими, яких люди тримали на руках. А овець на повідках визначав як собак. Подібні труднощі виникали й із козами: коли вони залазили на дерева — а вони іноді так роблять, — алгоритм вважав їх жирафами, тоді як інший схожий алгоритм називав птахами.

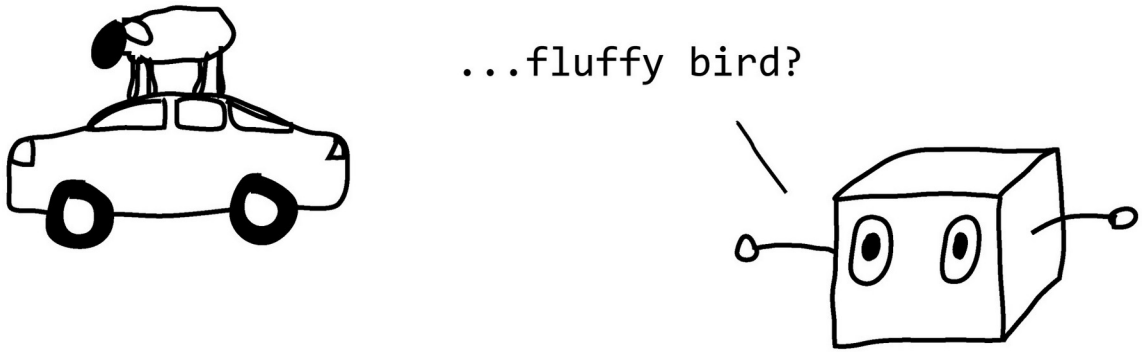
A herd of sheep grazing on
a lush green landscape



A herd of sheep grazing on
a lush green landscape



Я не могла знати напевно, але припустила, що ШІ створив правила на кшталт «зелена трава = вівця» і «хутро в автомобілях або на кухнях = кіт». Під час навчання вони добре працювали, але зазнали краху після зустрічі з реальним світом і запаморочливим різноманіттям ситуацій, у яких можуть опинитися вівці.



Система виявлення раку, яка насправді шукала лінійки

Такі помилки навчання часто трапляються в системах розпізнавання зображень. Але їхні наслідки можуть бути серйозними. Команда Стенфордського університету одного разу навчила ШІ відрізняти фотографії здорової шкіри від зображень раку шкіри. Однак після навчання дослідники виявили, що ненавмисно створили детектор лінійок: на багатьох фотографіях пухлин із навчального набору поруч лежали лінійки для позначення масштабу.[2]

ЯК ВИЯВИТИ ПОГАНЕ ПРАВИЛО

Чому внутрішні правила ШІ непрозорі

Помилки ШІ часто важко помітити. Ми не записуємо його правил: він вигадує їх сам, не занотовує й не пояснює так, як це зробила б людина. Натомість ШІ виконує складні взаємозалежні зміни у власній внутрішній структурі, перетворюючи універсальну основу на систему, тонко налаштовану для конкретного завдання. Це наче почати з кухні, повної звичайних інгредієнтів, а закінчити печивом. Правила можуть зберігатися у зв’язках між віртуальними клітинами мозку або в генах віртуального організму. Вони можуть бути складними, розпорошеними й дивно переплетеними між собою. Вивчення внутрішньої будови ШІ часто нагадує дослідження мозку або екосистеми — а щоб усвідомити їхню складність, не обов’язково бути нейробіологом чи екологом.

Дослідники намагаються з’ясувати, як саме ШІ ухвалює рішення, але загалом відкрити його внутрішні правила важко. Часто вони просто надто складні для розуміння. А інколи — особливо в комерційних або державних алгоритмах — сама система є закритою власністю. Тому проблеми нерідко стають помітними лише в результатах уже запущеного алгоритму, коли він ухвалює рішення, здатні впливати на життя й завдавати справжньої шкоди.

Як ШІ копіює людські упередження

Наприклад, ШІ, який рекомендував, кого з ув’язнених можна умовно-достроково звільнити, ухвалював упереджені рішення, несвідомо копіюючи расистську поведінку з навчальних даних.[3] Навіть не розуміючи, що таке упередження, ШІ все одно здатен бути упередженим. Зрештою, багато систем навчаються, наслідуючи людей. Вони відповідають не на запитання «Яке рішення найкраще?», а на запитання «Що зробили б люди?».

Систематичне тестування на упередженість допомагає виявляти деякі поширені проблеми ще до того, як вони завдадуть шкоди. Інша частина розв’язання — навчитися передбачати труднощі й заздалегідь проєктувати ШІ так, щоб їх уникати.

ЧОТИРИ ОЗНАКИ МАЙБУТНЬОЇ КАТАСТРОФИ ШІ

Як насправді може початися катастрофа ШІ

Коли люди уявляють катастрофу за участю ШІ, вони думають про системи, що відмовляються виконувати накази, вирішують, ніби в їхніх інтересах убити всіх людей, або створюють роботів-термінаторів. Але всі ці сценарії передбачають рівень критичного мислення й людиноподібного розуміння світу, якого ШІ не матиме в осяжному майбутньому. Як висловився провідний дослідник машинного навчання Ендрю Ин, боятися захоплення влади ШІ — це те саме, що непокоїтися через перенаселення Марса.[4]

Це не означає, що сучасний ШІ не здатен спричиняти проблеми. Від легкого роздратування програмістів до відтворення упереджень або аварії безпілотного автомобіля — нинішні системи не такі вже й нешкідливі. Однак базові знання про ШІ дають змогу заздалегідь помітити деякі загрози.

Ось як катастрофа ШІ могла б відбуватися вже сьогодні.

Уявімо стартап із Кремнієвої долини, який пропонує заощадити час компаній: відбирати кандидатів на роботу й визначати ймовірно найкращих працівників, аналізуючи короткі відеоспівбесіди. Пропозиція приваблива — компанії витрачають багато часу й ресурсів на розмови з десятками кандидатів, щоб знайти одну відповідну людину. Програма не втомлюється, не стає злою від голоду й не має особистих образ. Та за якими ознаками можна зрозуміти, що компанія насправді створює катастрофічний ШІ?

Ознака № 1: завдання надто складне

Добирати хороших працівників справді важко. Навіть люди не завжди впізнають сильних кандидатів. Ця людина щиро прагне тут працювати чи лише добре грає роль? Чи врахували ми інвалідність і культурні відмінності? Після додавання ШІ все стає ще складнішим. Йому майже неможливо зрозуміти нюанси жартів, інтонації та культурних відсилань. А якщо кандидат згадає поточні події? Коли ШІ навчався на даних попереднього року, він не зможе зрозуміти посилання й може покарати кандидата за нібито безглузду фразу. Для якісної роботи ШІ знадобиться величезний набір умінь і здатність стежити за великою кількістю інформації. Якщо він на це не здатен, на нас чекає невдача.

Ознака № 2: справжнє завдання не таке, як ми думали

Проблема проєктування ШІ для відбору кандидатів полягає в тому, що насправді ми просимо його не знайти найкращих людей. Ми просимо визначити кандидатів, найбільш схожих на тих, кого в минулому вподобали наші менеджери з наймання.

Можливо, це було б прийнятно, якби менеджери завжди ухвалювали чудові рішення. Але більшість американських компаній мають проблеми з різноманітністю — особливо серед керівників і в способах оцінювання резюме та кандидатів. За однакових інших умов резюме з іменами, що звучать як імена білих чоловіків, частіше приводять до співбесіди, ніж резюме з жіночими іменами та/або іменами представників меншин.[5] Навіть менеджери, які самі є жінками та/або належать до меншин, схильні несвідомо віддавати перевагу білим чоловікам.

Чимало поганих або відверто шкідливих систем ШІ створили люди, переконані, що проєктують засіб розв’язання певної проблеми, хоча несвідомо навчали його виконувати зовсім інше завдання.

Ознака № 3: існують підступні короткі шляхи

Пам’ятаєте ШІ для виявлення раку шкіри, який насправді був детектором лінійок? Розрізняти найменші відмінності між здоровими й раковими клітинами важко, тому системі було значно простіше перевіряти, чи є на фотографії лінійка.

Якщо ШІ для відбору кандидатів отримує упереджені навчальні дані — а це майже неминуче, якщо ви не доклали великих зусиль для їх очищення, — то разом із ними він дістає зручний короткий шлях до точнішого передбачення

«найкращого» кандидата: віддавати перевагу білим чоловікам. Це значно легше, ніж аналізувати нюанси добору слів. Або ШІ може знайти й використати інший невдалий спосіб. Наприклад, усіх успішних кандидатів знімали однією камерою, тож система навчиться читати метадані й відбиратиме лише тих, кого записали саме на неї.

ШІ постійно знаходить підступні короткі шляхи — він просто не розуміє, що так робити не слід!

Ознака № 4: ШІ навчався на недосконалих даних

У комп’ютерних науках є давній вислів: «сміття на вході — сміття на виході». Якщо мета ШІ — наслідувати людей, які ухвалюють недосконалі рішення, то ідеальним успіхом стане точне наслідування всіх цих рішень разом із помилками.

Хибні дані — чи то погані приклади для навчання, чи то недосконала симуляція з дивною фізикою — зіб’ють ШІ з пантелику або спрямують не туди. Оскільки в багатьох випадках наші приклади *і є* завданням, яке ми даємо ШІ, не дивно, що погані дані приводять до поганого розв’язання. Насправді ознаки з першої по третю найчастіше свідчать саме про проблеми з даними.

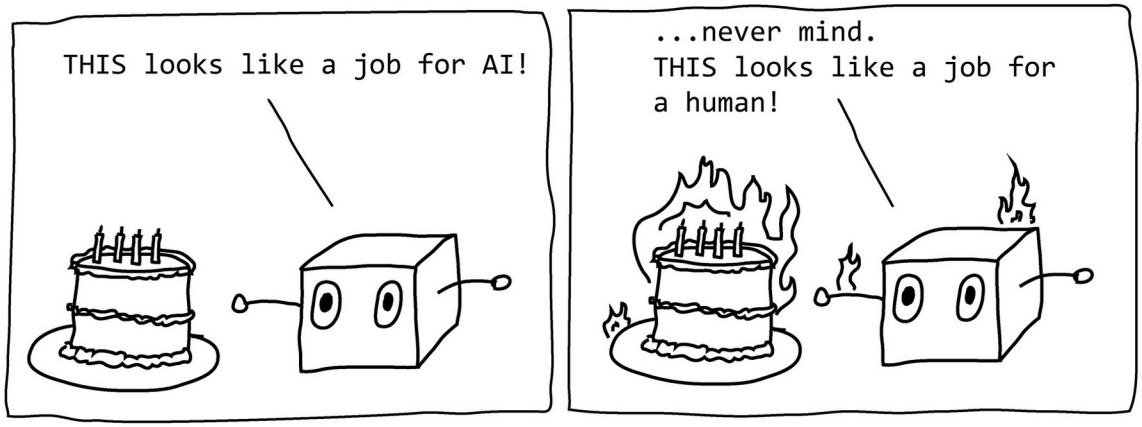
КАТАСТРОФА — ЧИ ЗАХВАТ

Успіх залежить від придатності завдання для ШІ

Приклад із відбором кандидатів, на жаль, не вигаданий. Кілька компаній уже пропонують послуги аналізу резюме або відеоспівбесід на основі ШІ, але лише деякі повідомляють, як борються з упередженнями, враховують інвалідність і культурні відмінності або з’ясовують, яку саме інформацію використовує їхня система. За ретельної роботи принаймні можливо створити ШІ для відбору кандидатів, який буде вимірювано менш упередженим, ніж люди-менеджери. Проте без оприлюдненої статистики, що це підтверджує, слід припускати, що упередженість нікуди не зникла.

Різниця між успішним і невдалим застосуванням ШІ зазвичай значною мірою залежить від того, наскільки завдання підходить для такого способу розв’язання. Існує чимало завдань, у яких ШІ ефективніший за людей. Що це за завдання й чому ШІ так добре з ними справляється? Погляньмо.

РОЗДІЛ 2. ШІ всюди, але де саме?



ЦЕЙ ПРИКЛАД СПРАВЖНІЙ, Я НЕ ЖАРТУЮ

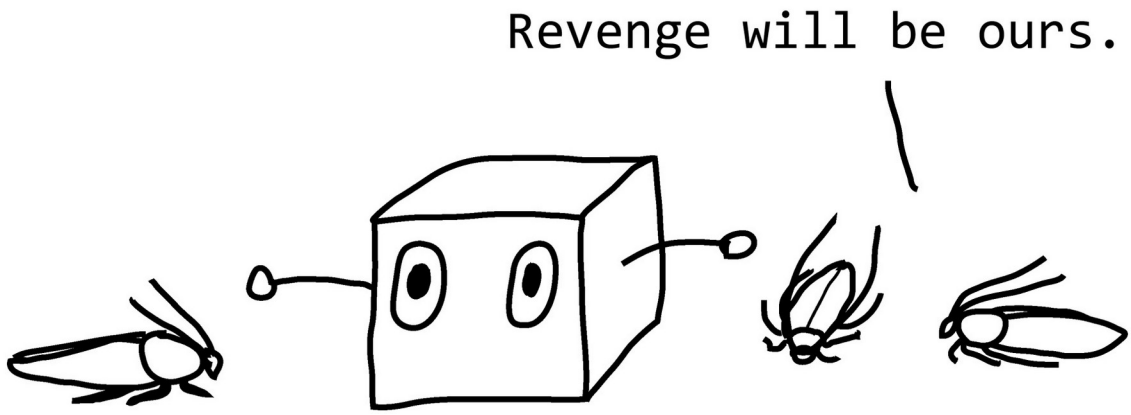
Ферма, що вирощує шість мільярдів тарганів

У китайському місті Січан є ферма, незвичайна одразу з кількох причин. По-перше, це найбільша ферма такого типу у світі з неперевершеною продуктивністю. Щороку вона вирощує шість мільярдів *Periplaneta americana* — понад двадцять вісім тисяч особин на кожен квадратний фут.[1] Щоб максимально підвищити продуктивність, ферма покладається на алгоритми: вони керують температурою, вологістю та постачанням корму, а також аналізують генетику й темпи росту *Periplaneta americana*.

Та головна дивина полягає в тому, що *Periplaneta americana* — це лише латинська назва звичайного американського таргана. Так, ферма справді вирощує тарганів. Потім їх подрібнюють і виготовляють мікстуру, яку високо цінують у традиційній китайській медицині. На пакованні зазначено: «Трохи солодка». Із «легким рибним запахом».

Оскільки точний принцип роботи алгоритму, що максимізує кількість тарганів, є цінною комерційною таємницею, подробиць відомо мало. Однак усе це дуже нагадує відомий уявний експеримент під назвою «максимізатор скріпок». У ньому надрозумний ШІ має єдине завдання — виготовляти канцелярські скріпки. Маючи таку одну-єдину мету, надрозумний ШІ *може* вирішити перетворити на скріпки всі доступні ресурси — навіть усю планету разом із її

мешканцями. На щастя — *на превелике* щастя, зважаючи на те, що ми щойно говорили про алгоритм, завдання якого полягає в максимізації кількості тарганів, — сучасним алгоритмам ще неймовірно далеко до самостійного керування фабриками чи фермами, не кажучи вже про перетворення світової економіки на машину з виробництва тарганів. Найімовірніше, тарганячий ШІ передбачає майбутні обсяги виробництва на основі минулих даних, а потім добирає умови середовища, які, на його думку, дадуть найбільше тарганів. Імовірно, він може пропонувати зміни в межах, установлених людьми-інженерами, але для збирання даних, виконання замовлень, розвантаження припасів і вкрай важливого маркетингу тарганячого екстракту все одно покладається на людей.



Чому ШІ добре підходить для оптимізації ферми

Та допомога в оптимізації тарганячої ферми — саме те завдання, яке ШІ, найімовірніше, виконуватиме добре. Даних дуже багато, а такі алгоритми вміють знаходити тенденції у величезних наборах. Робота навряд чи буде популярною серед людей, зате ШІ не дратують ані повторювані завдання, ані шурхіт мільйонів тарганячих лапок у темряві. Таргани швидко розмножуються, тож наслідки зміни параметрів стають помітними доволі скоро. До того ж це конкретне вузьке завдання, а не складна проблема з невизначеними межами.

Коли оптимізація відчиняє тарганам двері

Чи може застосування ШІ для максимізації кількості тарганів усе одно спричинити проблеми? Так. ШІ не розуміє загального контексту: чого саме й навіщо насправді намагається досягти. Тому він часто розв’язує завдання несподіваними способами. Припустімо, тарганячий ШІ виявив, що, виставивши і нагрівання, і подачу води в певному приміщенні на максимум, можна значно збільшити кількість вирощених там тарганів. Він не матиме способу дізнатися

— та йому буде байдуже, — що насправді спричинив коротке замикання у дверях, які не давали тарганам потрапити на кухню працівників.

Формально коротке замикання у дверях означає, що ШІ добре виконав свою роботу. Його завданням було збільшити кількість тарганів, а не завадити їм утекти. Щоб ефективно працювати зі ШІ й передбачати неприємності, перш ніж вони трапляться, потрібно розуміти, для чого машинне навчання підходить найкраще.

ВЛАСНЕ, Я НЕ ПРОТИ, ЩОБ РОБОТ ЗАБРАВ ЦЮ РОБОТУ

Автоматизація масових і повторюваних завдань

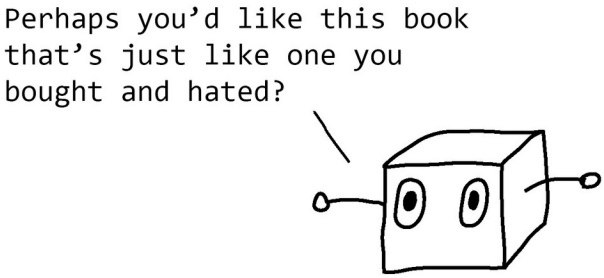
Алгоритми машинного навчання корисні навіть у роботі, яку людина виконала б краще. Застосування алгоритму позбавляє витрат і клопоту, пов’язаних із залученням людини, особливо коли завдання масове й повторюване. Звісно, це стосується не лише машинного навчання, а автоматизації загалом. Якщо робот-пилосос Roomba звільняє нас від потреби самостійно пилососити кімнату, ми терпітимемо те, що його знову й знову доводиться витягати з-під дивана.

Медичні зображення, спам і гіперперсоналізація

Одне з повторюваних завдань, яке люди автоматизують за допомогою ШІ, — аналіз медичних зображень. Лаборанти щодня годинами розглядають зразки крові під мікроскопом, підраховують тромбоцити, лейкоцити й еритроцити або перевіряють зразки тканин на аномальні клітини. Кожне з цих завдань просте, послідовне й самодостатнє, а отже добре підходить для автоматизації. Проте ставки зростають, коли алгоритми залишають дослідницьку лабораторію й починають працювати в лікарнях, де наслідки помилки значно серйозніші. Схожа проблема виникає з безпілотними автомобілями: керування здебільшого повторюване, і було б добре мати водія, який ніколи не втомлюється, але навіть крихітний збій на швидкості сто кілометрів на годину може спричинити тяжкі наслідки.

Інше масове завдання, яке ми охоче автоматизуємо, навіть якщо ШІ ще не досяг людського рівня, — фільтрування спаму. Потік спаму мінливий і сповнений нюансів, тому для ШІ це непроста проблема. З іншого боку, більшість із нас погодиться час від часу знаходити лист не в тій папці, якщо загалом вхідні повідомлення залишатимуться чистими. Позначення шкідливих вебадрес, фільтрування дописів у соціальних мережах і виявлення ботів — масові завдання, у яких ми переважно терпимо недосконалу роботу.

Ще одна сфера, де ШІ починає демонструвати користь, — гіперперсоналізація. Компанії використовують його для індивідуальних рекомендацій товарів, фільмів і музичних добірок, пристосовуючи досвід до кожного споживача так, як було б надто дорого робити вручну. То й що, якщо ШІ переконаний, що нам потрібна нескінченна кількість килимків для коридору, або вважає нас малюком, бо одного разу ми купили подарунок до народження дитини? Його помилки здебільшого нешкідливі — крім випадків, коли вони *дуже-дуже* невдалі, — зате можуть принести компанії продаж.



Новини з таблиці: журналістський алгоритм Heliograf

Комерційні алгоритми вже здатні писати гіперлокальні статті про результати виборів, рахунки спортивних матчів і нещодавні продажі будинків. У кожному випадку алгоритм створює лише дуже шаблонний текст, але людей настільки цікавить сама інформація, що це, схоже, не заважає. Один із таких алгоритмів має назву Heliograf. Його розробила газета *Washington Post*, щоб перетворювати спортивну статистику на новинні статті. Уже 2016 року він створював сотні матеріалів на рік. Ось приклад його репортажу про матч з американського футболу.[2]

> У п'ятницю Quince Orchard Cougars розгромили Einstein Titans із рахунком 47:0.
>
> Quince Orchard відкрили рахунок восьмиярдовим тачдауном після заблокованого панта у виконанні Аарона Ґріна. Cougars збільшили перевагу завдяки триярдовому тачдаун-ривку Маркеза Купера. Потім Cougars ще більше відірвалися завдяки вісімнадцятиярдовому тачдаун-ривку Аарона Дервіна. Відрив зріс після того, як Дервін прийняв від квотербека Дока Боннера передачу на шістдесят три ярди й виконав тачдаун, довівши рахунок до 27:0.

Текст не надто захопливий, але Heliograf справді описує гру.[*] Він уміє заповнювати статтю на основі таблиці з даними та кількох стандартних спортивних фраз. Проте ШІ на зразок Heliograf цілковито провалився б, зіткнувшись з інформацією, яка не вкладається в передбачені клітинки. Чи вибіг посеред матчу на поле кінь? Чи заполонили таргани роздягальню

Einstein Titans? Чи можна тут дотепно пожартувати? Heliograf уміє повідомляти лише те, що є в його таблиці.

Утім, тексти, створені ШІ, дають новинним виданням змогу випускати матеріали, які раніше були економічно невигідними. Людина має вирішити, які статті автоматизувати, а також створити основні шаблони й типові фрази. Та щойно газета налаштує такий вузькоспеціалізований алгоритм, він зможе виробляти стільки новин, скільки знайдеться таблиць із даними. Наприклад, один шведський новинний сайт створив Homeowners Bot — «бота домовласників», який читав таблиці з даними про нерухомість і перетворював кожен продаж на окрему статтю. За чотири місяці він написав понад десять тисяч матеріалів. Саме цей тип статей став на сайті найпопулярнішим і найприбутковішим.[3] А журналісти-люди натомість можуть витрачати свій цінний час на творчу розслідувальну роботу. Великі новинні видання дедалі частіше застосовують ШІ під час написання статей.[4]

Астрономія, творчі інструменти та дипфейки

Наука — ще одна сфера, у якій ШІ перспективний для автоматизації повторюваних завдань. Наприклад, фізики використовували ШІ, щоб спостерігати за світлом далеких зірок[5] і шукати характерні ознаки наявності планети. Звісно, ШІ поступався точністю фізикам, які його навчали. Більшість позначених ним «цікавих» зірок виявилися хибними тривогами. Однак він правильно відкинув як *нецікаві* понад 90 відсотків зірок, заощадивши науковцям багато часу.

З’ясувалося, що в астрономії повно величезних наборів даних. За весь час роботи телескоп Euclid збере десятки мільярдів зображень галактик, і, можливо, лише двісті тисяч із них міститимуть свідчення гравітаційного лінзування.[6] Воно виникає, коли надмасивна галактика має настільки сильну гравітацію, що викривляє світло інших, віддаленіших галактик. Знайшовши такі лінзи, астрономи зможуть багато дізнатися про гравітацію в колосальному міжгалактичному масштабі, де залишається стільки загадок, що походження цілих 95 відсотків маси й енергії Всесвіту досі не пояснено. Переглядаючи зображення, алгоритми працювали швидше за людей, а іноді навіть точніше. Проте коли телескоп зафіксував одну надзвичайно цікаву «джекпот-лінзу», помітили її лише люди.

Творчу роботу також можна частково автоматизувати — принаймні під наглядом митця. Раніше фотограф міг годинами коригувати знімок, а сучасні фільтри на основі ШІ, зокрема вбудовані в Instagram і Facebook, непогано

налаштовують контраст та освітлення й навіть додають ефект глибини різкості, імітуючи дорогий об’єктив. Більше не потрібно вручну домальовувати другові котячі вуха: фільтр Instagram сам визначить, де їх розмістити, навіть коли людина рухає головою. У великих і малих справах ШІ дає художникам і музикантам інструменти, що заощаджують час і розширюють можливості самостійної творчості. Зворотний бік — інструменти створення **дипфейків**, які дають змогу замінити голову та/або тіло однієї людини на іншу навіть у відео. З одного боку, доступність такого засобу дозволяє митцям вставляти Ніколаса Кейджа чи Джона Чо в різні кіноролі — заради жарту або щоб серйозно порушити питання представлення меншин у Голлівуді.[7] З іншого — дедалі простіше створення дипфейків уже дає переслідувачам нові способи виготовляти тривожні, спрямовані проти конкретних людей відео й поширювати їх в інтернеті. Технологія вдосконалюється, підробки стають переконливішими, і людей та уряди непокоїть можливість створювати фальшиві, але руйнівні ролики — наприклад, реалістичне відео, у якому політик нібито виголошує провокаційну заяву.

Послідовність не означає неупередженість

Автоматизація на основі ШІ не лише заощаджує час людей, а й може забезпечити послідовніші результати. Ефективність окремої людини змінюється протягом дня залежно від того, коли вона востаннє їла, скільки спала, який у неї настрій і які упередження. Безліч досліджень показали, що сексизм, расові упередження, ейблізм та інші проблеми впливають на те, чи потрапить резюме до короткого списку, чи отримає працівник підвищення зарплати та чи звільнять ув’язненого умовно-достроково. Алгоритми уникають людської непослідовності: отримавши певний набір даних, вони видадуть майже однаковий результат незалежно від того, ранок зараз, полудень чи «щаслива година» в барі. На жаль, послідовність не дорівнює неупередженості. Алгоритм цілком може бути послідовно несправедливим — особливо якщо, як багато систем ШІ, навчався, копіюючи людей.

Отже, існує чимало завдань, які привабливо автоматизувати за допомогою ШІ. Але від чого залежить, чи *можемо* ми автоматизувати певну проблему?

ЩО ВУЖЧЕ ЗАВДАННЯ, ТО РОЗУМНІШИМ ЗДАЄТЬСЯ ШІ

Тест Тюрінга можна обійти вузькою темою

Тест Тюрінга став відомим мірилом інтелекту комп’ютерної програми відтоді, як Алан Тюрінг запропонував його у 1950-х роках. Програма проходить

стандартний тест Тюрінга, якщо здатна спілкуватися з людьми й переконати приблизно третину з них, що перед ними людина, а не комп’ютер. Проходження тесту інколи вважають ознакою того, що алгоритм досяг людського рівня інтелекту й, можливо, навіть самосвідомості. Сюжети багатьох науково-фантастичних книжок і фільмів — *«Той, хто біжить по лезу», «Ex Machina», «Двохсотрічна людина»* та інших — розповідають про розвинений загальний штучний інтелект, який довів свою «особистість», пройшовши тест Тюрінга.

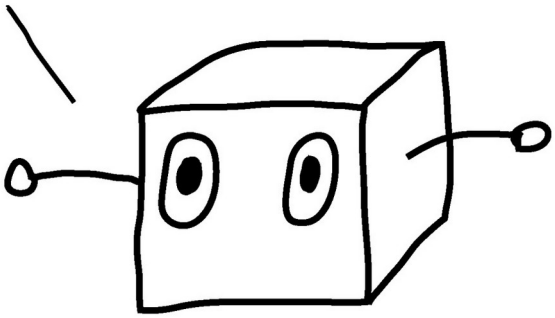
Насправді тест Тюрінга погано вимірює інтелект алгоритму. По-перше, пройти його легко, якщо достатньо звузити тему розмови. Щоб перевірити цю думку, я поспілкувалася у Facebook із ботом мережі Whole Foods Market:

> **Whole Foods:** Привіт, Джанель! Ми допоможемо знайти рецепти, такі ж прості, як їхні інгредієнти.
> **Я:** У вас є рецепт гуакамоле?
> **Whole Foods:** <публікує рецепт: гуакамоле з кінзою>
> **Я:** Чи можна приготувати гуакамоле із зеленого горошку?
> **Whole Foods:** <публікує рецепт: гуакамоле із зеленого горошку>
> **Я:** Дай мені рецепт з авокадо. Будь-що, крім гуакамоле.
> **Whole Foods:** <публікує рецепт: сальса з авокадо, оливками й кінзою>

На цьому етапі ситуація трохи неоднозначна. Мій співрозмовник не сказав прямо, людина він чи бот, і вправно відповідає на запитання, для яких достатньо опублікувати рецепт. Але щойно я відхиляюся від вузького формату, бот швидко себе викриває.

> **Я:** Як визначити, чи стигле авокадо?
> **Whole Foods:** <публікує рецепт: гуакамоле з едамаме>
> **Я:** У тебе є улюблений персонаж «Зоряних війн»?
> **Whole Foods:** <публікує рецепт: обсмажені креветки з полентою й сиром манчего>

I *loved* Sautéed Shrimp with Polenta and Manchego in The Last Jedi!

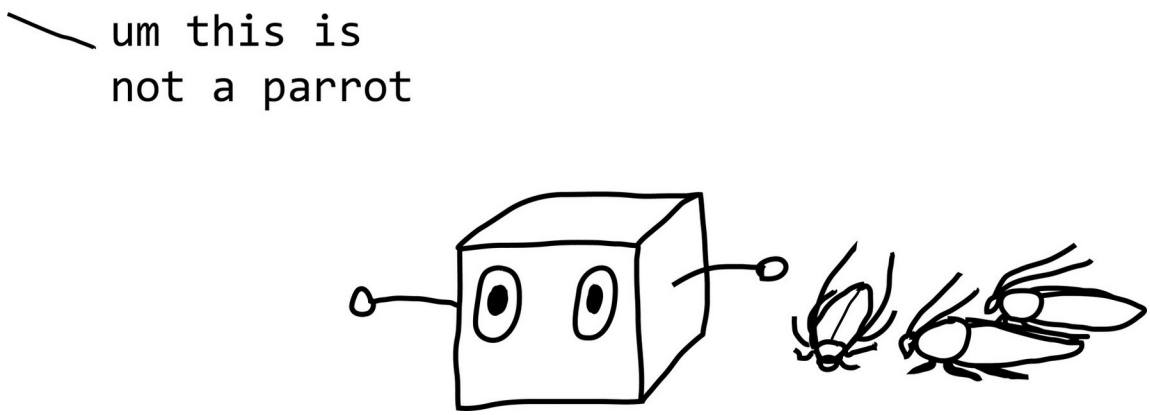


Як людська ввічливість допомагає чатботам

Компанії застосовують цю стратегію, коли хочуть залучити чатботів до обслуговування клієнтів. Вони не називають ботів ботами, а покладаються на людську ввічливість: вона втримує розмову в межах тем, де алгоритм здатен дати собі раду. Зрештою, якщо існує ймовірність, що ви спілкуєтеся з живим працівником, було б нечемно випробовувати його дивними запитаннями не за темою.

Чому Facebook M не став універсальним помічником

Навіть коли клієнти не відхиляються від визначеної теми, чатбот матиме труднощі, якщо вона надто широка. У серпні 2015 року Facebook почав створювати чатбота на основі ШІ під назвою M. Він мав бронювати готелі, замовляти квитки до театру, радити ресторани тощо.[8] Компанія планувала спочатку доручати найскладніші запити людям, накопичуючи велику кількість прикладів для навчання алгоритму. Згодом Facebook сподівався зібрати достатньо даних, щоб M самостійно відповідав на більшість запитань. На жаль, почувши, що M можна просити про будь-що, клієнти сприйняли ці слова буквально. Інженер, який започаткував проєкт, розповідав в інтерв’ю: «Спочатку люди запитують про погоду на завтра; потім кажуть: “Чи є вільний італійський ресторан?” Далі в них виникає запитання про імміграцію, а трохи згодом вони просять M організувати весілля».[9] Один користувач навіть попросив M організувати візит папуги до свого друга. M впорався — передавши запит людині. Через кілька років після запуску Facebook виявив, що алгоритму й досі потрібно надто багато людської допомоги. У січні 2018 року сервіс закрили.[10]



Робота з усім розмаїттям того, що людина може сказати або попросити, — надзвичайно широке завдання. Розумові можливості ШІ досі мізерні порівняно з людськими, і що ширшим стає завдання, то більше алгоритм безпорадний.

Вузьке й широке навчання на рецептах

Наприклад, нещодавно я навчала ШІ створювати рецепти. Цей конкретний алгоритм був налаштований наслідувати текст, але починав із чистого аркуша: не знав, що таке рецепти, не розумів, що різні літери позначають інгредієнти й дії з ними, і навіть не знав англійської мови. Потрібно було стежити за безліччю речей, але ШІ старанно намагався визначити, як розміщувати одну літеру після іншої й наслідувати побачені рецепти. Коли я дала йому для навчання лише рецепти тортів, він створив ось це:

> **Морквяний торт («торти Вера Лейдіс», алкоголь**
> 1 пакунок жовтої суміші для торта
> 3 склянки борошна
> 1 чайна ложка розпушувача
> 1 ½ чайної ложки харчової соди
> ¼ чайної ложки солі
> 1 чайна ложка меленої кориці
> 1 чайна ложка меленого імбиру
> ½ чайної ложки меленої гвоздики
> 1 чайна ложка розпушувача
> ½ чайної ложки солі
> 1 чайна ложка ванілі
> 1 яйце кімнатної температури
> 1 склянка цукру
> 1 чайна ложка ванілі
> 1 склянка подрібнених горіхів пекан
> Розігрійте духовку до 175 °C. Змастіть роз’ємну форму діаметром 23 см.
> Щоб приготувати торт: збивайте яйця на високій швидкості, доки вони не стануть густими й жовтими, та відкладіть. В окремій мисці збийте білки до

> стійких піків. Швидкістю перше як суміш у підготовлену форму та розрівняйте тісто. Випікайте приблизно 40 хвилин або доки дерев’яна зубочистка, вставлена в центр, не вийде чистою. Остудіть у формі 10 хвилин. Перекладіть на решітку й повністю остудіть.
> Вийміть торт із форми, щоб повністю остудити. Подавайте теплим.
> Кулінарна книга HereCto (1989), з кухні та Hawn inthe Canadian Living
> Вихід: 16 порцій

Рецепт недосконалий, але принаймні в ньому можна впізнати торт. Хоча, уважно прочитавши інструкцію, ви зрозумієте, що в результаті отримаєте лише один запечений яєчний жовток.

Потім я попросила ШІ навчитися створювати не тільки торти, а й супи, барбекю, печиво та салати. Даних стало приблизно вдесятеро більше: 24 043 загальні рецепти замість лише 2431 рецепта торта. Ось що він створив:

> **Намазка з курки та рису, сир/яйця, салати, сир**
> 2 фунти сердець без насіння
> 1 склянка подрібненої свіжої м’яти або малинового пирога
> ½ склянки катрімасу, натертого
> 1 столова ложка рослинної олії
> 1 сіль
> 1 перець
> 2 ½ столової ложки цукру, цукру
> З’єднайте нелистя й перемішуйте, доки суміш не стане густою. Потім додайте яйця, цукор, мед і кмин та готуйте на слабкому вогні. Додайте кукурудзяний сироп, орегано, розмарин і білий перець. Вкладіть вершки теплом. Готуйте, додайте решту 1 чайної ложки розпушувача й сіль. Випікайте при 175 °C від 2 до 1 години. Подавайте гарячим.
> Вихід: 6 порцій

Цього разу рецепт став цілковитою катастрофою. ШІ довелося визначати, коли використовувати шоколад, а коли — картоплю. Деякі страви потрібно було випікати, інші — довго тушкувати, а салати взагалі не вимагали теплової обробки. Намагаючись засвоїти й утримати всі ці правила, ШІ надто розпорошив свої обмежені розумові можливості.

Тому люди, які навчають ШІ розв’язувати комерційні або дослідницькі завдання, зрозуміли: краще робити його вузьким спеціалістом. Якщо певний алгоритм здається розумнішим за ШІ, що вигадав «Намазку з курки та рису», головна відмінність, імовірно, полягає в тому, що він отримав вужче й краще дібране завдання. Що вужчий ШІ, то розумнішим він здається.

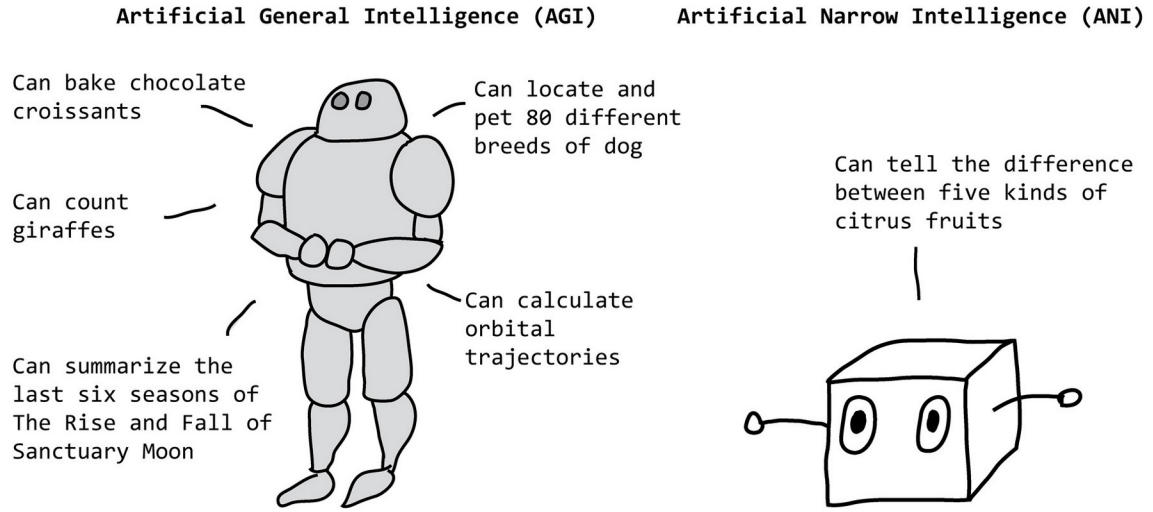
С-3PO ПРОТИ ВАШОГО ТОСТЕРА

Вузький ШІ та загальний штучний інтелект

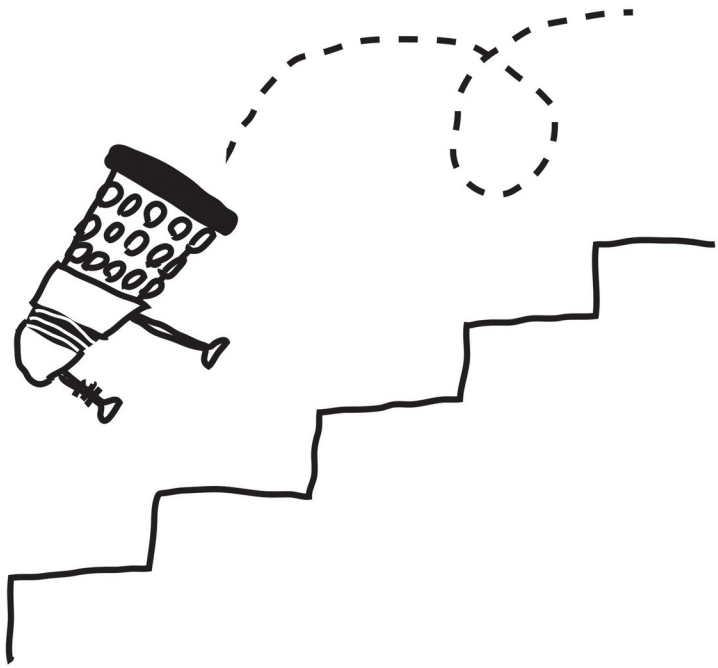
Саме тому дослідники розрізняють **вузький штучний інтелект (ВШІ, або ANI)**, який ми маємо нині, та **загальний штучний інтелект (ЗШІ, або AGI)**, знайомий нам переважно з книжок і фільмів. Ми звикли до історій про надрозумні комп'ютерні системи на кшталт Skynet і HAL або дуже людиноподібних роботів — WALL-E, С-3PO, Дейту та інших. ШІ в цих історіях може не розуміти тонкощів людських емоцій, але здатен осмислювати величезне розмаїття предметів і ситуацій та реагувати на них. ЗШІ міг би перемогти вас у шахах, розповісти історію, спекти торт, описати вівцю й назвати три речі, більші за омара. Однак він досі твердо належить до наукової фантастики. Більшість фахівців погоджується, що до появи ЗШІ залишаються багато десятиліть — якщо він узагалі колись стане реальністю.

Нинішній ВШІ значно простіший. *Набагато* простіший. Порівняно з С-3PO це, по суті, тостер.

Алгоритми, які потрапляють у заголовки новин, перемагаючи людей у шахах або ґо, перевершують нас лише в одному спеціалізованому завданні. Утім, машини вже давно кращі за людей у певних окремих справах. Калькулятор завжди перевершував людину в діленні багатозначних чисел — але досі не здатен спуститися сходами.



> До речі, чимало науково-фантастичних ЗШІ з якоїсь причини також не можуть спускатися сходами. Далеки, С-3PO, оте щось із *«Робокопа»*, HAL. Варто дослідити докладніше?



Чому розпізнавання емоцій не є вузьким завданням

Які проблеми достатньо вузькі для сучасних алгоритмів ВШІ? На жаль — згадайте ознаку № 1 майбутньої катастрофи ШІ: «завдання надто складне», — реальна проблема часто ширша, ніж здається спочатку. У прикладі з аналізом відеоспівбесід із розділу 1 завдання на перший погляд доволі вузьке: просто розпізнати емоції на людських обличчях. Але що робити з кандидатами, які перенесли інсульт, мають шрами на обличчі або виражають емоції не нейротиповим способом? Людина здатна зрозуміти ситуацію кандидата й відповідно скоригувати очікування. Щоб зробити те саме, ШІ мав би розпізнати сказані слова — а перетворення мовлення на текст саме собою є окремою проблемою ШІ, — зрозуміти їхній зміст — сучасні системи тлумачать лише обмежені типи речень у вузьких предметних сферах і погано працюють із нюансами, — а потім використати ці знання, щоб змінити інтерпретацію емоційних даних. Нинішній ШІ не здатен виконати настільки складне завдання й, найімовірніше, відсіяв би всіх цих людей ще до того, як їх побачить людина.

Як ми побачимо далі, безпілотне керування автомобілем може бути ще одним завданням, значно ширшим, ніж здається спочатку.

НЕДОСТАТНЬО ДАНИХ — ОБЧИСЛЕННЯ НЕМОЖЛИВЕ

ШІ потребує тисяч і мільйонів прикладів

ШІ навчається повільно. Якби людині показали фотографію нової тварини під назвою «вуґ», а потім дали велику добірку зображень і попросили знайти всі

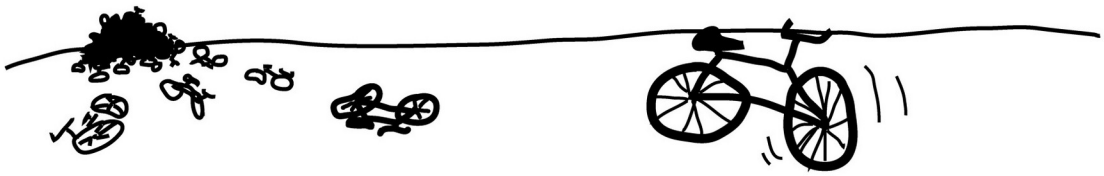
фотографії з вуґами, вона, ймовірно, непогано впоралася б, побачивши лише один приклад. Натомість ШІ можуть знадобитися тисячі або сотні тисяч фотографій, перш ніж він хоча б відносно надійно навчиться впізнавати вуґів. До того ж зображення мають бути достатньо різноманітними, щоб алгоритм зрозумів: слово «вуґ» позначає тварину, а не картату підлогу, на якій вона стоїть, чи людську руку, що гладить її по голові.

Дослідники працюють над системами, здатними опановувати тему за меншою кількістю прикладів. Цю властивість називають **навчанням на одному прикладі**, або **one-shot learning**. Але поки що для розв’язання проблеми за допомогою ШІ потрібні гори навчальних даних. Популярний набір ImageNet для навчання створення й розпізнавання зображень містить 14 197 122 зображення лише у тисячі різних категорій. Так само людині може вистачити кількох сотень годин практики, перш ніж їй дозволять водити автомобіль самостійно, тоді як до 2018 року безпілотні автомобілі компанії Waymo зібрали дані з понад шести мільйонів миль реальних доріг і ще п’яти мільярдів миль у симуляціях.[11] І до масового впровадження безпілотних автомобілів тоді все одно було далеко. Ненаситність ШІ до даних — одна з головних причин, чому епоха «великих даних», коли люди збирають і аналізують величезні масиви інформації, йде пліч-о-пліч з епохою ШІ.

Прискорене навчання в симуляціях

Іноді ШІ навчається настільки повільно, що робити це в реальному часі непрактично. Тоді навчання прискорюють, уміщуючи сотні років досвіду в кілька годин. Програма OpenAI Five навчалася грати в комп’ютерну гру Dota — мережеву фентезі-гру, де команди мають спільно захопити карту. Вона змогла перемогти одних із найсильніших гравців світу, граючи не проти людей, а проти самої себе. ШІ одночасно проводив десятки тисяч партій і щодоби накопичував досвід, еквівалентний 180 рокам гри.[12] Навіть коли кінцева мета пов’язана з реальним світом, інколи варто побудувати симуляцію, щоб заощадити час і зусилля.

Інший ШІ мав навчитися втримувати рівновагу на велосипеді. Проте й він виявився повільним учнем. Програмісти записували всі траєкторії переднього колеса, поки велосипед знову й знову хитався та падав. Понад сто аварій знадобилося, щоб ШІ зміг проїхати без падіння хоча б кілька метрів, і ще тисячі — щоб подолати кілька десятків метрів.



Ризик навчання у спрощеному світі

Навчати ШІ в симуляції зручно, але ризиковано. Через обмежену обчислювальну потужність комп'ютерів симуляції далеко не такі докладні, як реальний світ, і неминуче тримаються на всіляких спрощеннях та обхідних рішеннях. Це стає проблемою, якщо ШІ помічає такі скорочення й починає їх використовувати. Докладніше про це поговоримо далі.

ЯК СКОРИСТАТИСЯ ЧУЖИМ ПРОГРЕСОМ

Перенесення навчання: від метал-гуртів до морозива

Навіть без величезного набору даних проблему інколи можна розв'язати за допомогою ШІ, якщо ви або хтось інший уже виконав схоже завдання. Коли ШІ починає не з нуля, а з конфігурації, засвоєної на попередньому наборі даних, він може повторно використати чимало знань. Припустімо, я вже навчила ШІ вигадувати назви метал-гуртів. Наступне завдання — побудувати ШІ, що створює смаки морозива. Я можу швидше отримати результат і потребуватиму менше прикладів, якщо почну з моделі метал-гуртів. Адже, навчаючись вигадувати їхні назви, ШІ вже з'ясував:

- приблизну довжину кожної назви;
- що першу літеру кожного рядка слід писати великою;
- поширені сполучення літер — *ch, va, str* і *pis* (він уже частково навчився писати *chocolate, vanilla, strawberry* та *pistachio* — «шоколад, ваніль, полуниця й фісташка»!);
- поширені слова на кшталт *the* і, гм… *death* — «смерть»?

Тож кілька коротких кіл навчання можуть перетворити модель, яка створює таке:

> Dragonred of Blood *[Криваво-червоний дракон]*
> Stäggabash
> Deathcrack *[Тріщина смерті]*

> Stormgarden *[Сад бурі]*
> Vermit
> Swiil
> Inbumblious
> Inhuman Sand *[Нелюдський пісок]*
> Dragonsulla and Steelgosh
> Chaosrug *[Килим хаосу]*
> Sespessstion Sanicilevus

на модель, яка створює ось це:

> Лимонний Oreo *(Lemon-Oreo)*
> Полуничний чурос *(Strawberry Churro)*
> Вишневий масала-чай *(Cherry Chai)*
> Солодове чорне божевілляяя *(Malted Black Madnesss)*
> Гарбузово-гранатовий шоколадний батончик *(Pumpkin Pomegranate Chocolate Bar)*
> Копчені какао-крупинки *(Smoked Cocoa Nibe)*
> Підсмажений базилік *(Toasted Basil)*
> Гірський інжир і полуничний вихор *(Mountain Fig n Strawberry Twist)*
> Шоколадна шоколадна шоколадна шоколадна дорога *(Chocolate Chocolate Chocolate Chocolate Road)*
> Шоколад арахіс шоколад шоколад шоколад *(Chocolate Peanut Chocolate Chocolate Chocolate)*

Між цими етапами є лише *зовсіім невелика* незручна фаза, коли ШІ створює такі назви:

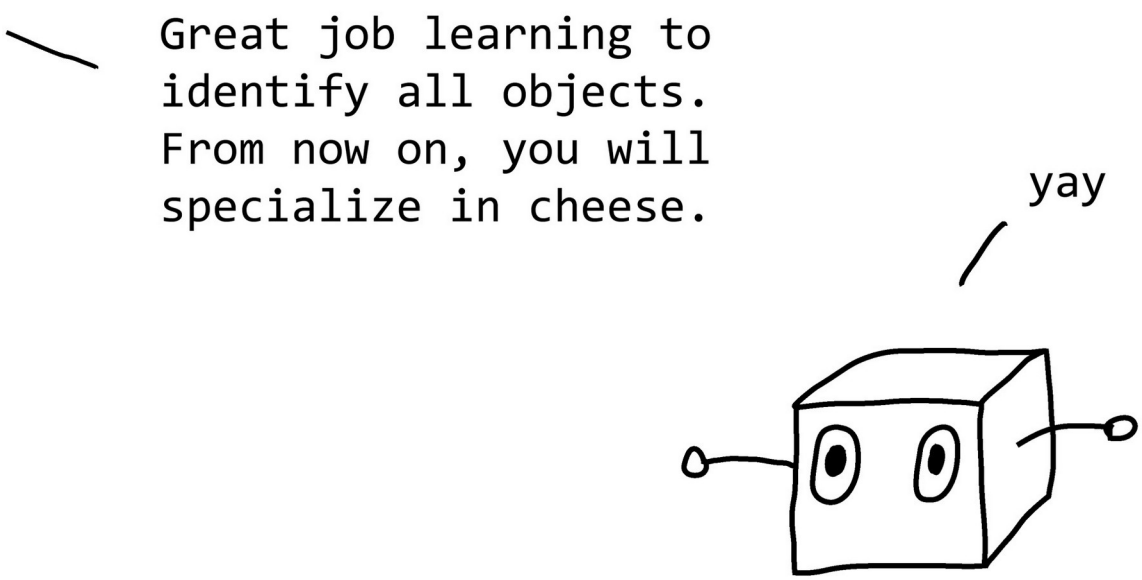
> Вихор пекла *(Swirl of Hell)*
> Людський крем *(Person Cream)*
> Нічношинка з ірисом *(Nightham Toffee)*
> Смерть Фітберрардерна *(Feethberrardern’s Death)*
> Некрозірка з шоколадною людиною *(Necrostar with Chocolate Person)*
> Жалобна пісня помадки *(Dirge of Fudge)*
> Звіриний крем *(Beast Cream)*
> Кінець усьому *(End All)*
> Сир смерті *(Death Cheese)*
> Кривавий пекан *(Blood Pecan)*
> Мовчання кокоса *(Silence of Coconut)*
> Масляний вогонь *(The Butterfire)*
> Павук і сорбозвір *(Spider and Sorbeast)*
> Ожиновий опік *(Blackberry Burn)*

Можливо, краще було почати з пирогів.

Виявляється, моделі ШІ повторно використовують дуже часто. Цей процес називають **перенесенням навчання**, або **transfer learning**. Починаючи із ШІ, який уже пройшов частину шляху до мети, можна обійтися меншою кількістю даних і заощадити багато часу. Навчання найскладніших алгоритмів на найбільших наборах даних навіть на дуже потужних комп’ютерах триває дні або тижні. А перенесення навчання дає змогу за хвилини чи секунди пристосувати той самий ШІ до схожого завдання.

Повторне використання моделей розпізнавання зображень

Особливо часто перенесення навчання використовують у розпізнаванні зображень, адже навчання нової системи з нуля потребує дуже багато часу й даних. Зазвичай беруть алгоритм, уже навчений розпізнавати звичайні предмети на загальних зображеннях, а потім використовують його як відправну точку для спеціалізованого розпізнавання. Наприклад, якщо система вже знає правила, що допомагають розпізнавати вантажівки, котів і футбольні м’ячі, вона має добрий початок для розрізнення овочів та фруктів у сканері супермаркету. Чимало правил, які відкриває універсальна модель, — як знаходити краї, визначати форми та класифікувати текстури — стануть у пригоді й такому сканеру.



НЕ ПРОСІТЬ ЙОГО ПАМ’ЯТАТИ

Короткозорий ШІ у відеоіграх

Проблему легше розв’язати за допомогою ШІ, якщо вона майже не потребує пам’яті. Через обмежені розумові можливості системи особливо погано запам’ятовують. Це добре помітно, коли ШІ грає у відеоігри. Він марнотратно витрачає життя персонажа й інші ресурси — наприклад, потужні атаки, кількість яких обмежена. Спочатку ШІ бездумно витрачає життя й закляття, доки їх не залишається критично мало, а тоді раптово починає діяти обережно.[13]

Один ШІ навчився грати в *Karate Kid*, але завжди витрачав усі потужні прийоми «удар журавля» на самому початку. Чому? Йому вистачало пам’яті, щоб заглядати лише на шість секунд уперед. Том Мерфі, який навчав алгоритм, пояснив це так: «Якщо щось знадобиться через шість секунд — ну, не пощастило. Марнування життів та інших ресурсів — поширений тип невдачі».[14]

Навіть такі складні алгоритми, як бот OpenAI для Dota, мають обмежене часове вікно пам’яті й передбачення. OpenAI Five здатен планувати на разючі дві хвилини вперед — це справді багато для гри, де швидко відбувається безліч складних подій, — але матчі Dota можуть тривати сорок п’ять хвилин і довше. OpenAI Five грає з моторошною агресивністю й точністю, проте, схоже, не вміє застосовувати стратегії, користь від яких з’явиться лише в далекій перспективі.[15] Подібно до простого бота Karate Kid, що надто рано

використовує удар журавля, він схильний витрачати найпотужніші атаки персонажа на початку, а не берегти до моменту, коли вони матимуть найбільше значення.

Чому алгоритм не може злізти з уступу

Нездатність планувати наперед трапляється доволі часто. На другому рівні *Super Mario Bros.* є сумнозвісний уступ — прокляття всіх ігрових алгоритмів. На ньому лежить багато блискучих монет! До другого рівня ШІ зазвичай уже знає, що монети — це добре. Також розуміє, що треба постійно рухатися праворуч, аби встигнути дістатися кінця рівня. Але якщо стрибнути на уступ, потім доведеться трохи піти назад, щоб зістрибнути. Раніше ШІ ніколи не був змушений рухатися назад. Він не може цього збагнути й застрягає, доки не закінчиться час. «Я буквально витратив на цю проблему приблизно шість вихідних і тисячі годин процесорного часу», — розповідав Том Мерфі. Зрештою він подолав уступ, поліпшивши здатність свого ШІ до довгострокового планування.[16]

Коротка пам’ять у перекладі й створенні тексту

Коротка пам’ять ШІ створює проблеми й під час генерування тексту. Журналістський алгоритм Heliograf може перетворювати окремі рядки таблиці на речення шаблонного спортивного матеріалу саме тому, що пише кожне речення майже незалежно. Йому не потрібно одночасно пам’ятати всю статтю.

Нейромережі для перекладу мов, подібні до тих, що лежать в основі Google Translate, також не мають пам’ятати цілі абзаци. Зазвичай речення або навіть його частину можна перекласти окремо, не згадуючи попереднього речення. Коли виникає довгострокова залежність — наприклад, неоднозначність, яку можна було б усунути за допомогою інформації з попереднього речення, — ШІ зазвичай не здатен нею скористатися.

В інших завданнях жахлива пам’ять ШІ стає ще помітнішою. Один із прикладів — алгоритмічно створені оповідання. Не випадково ШІ не пише книжок або телесеріалів, хоча люди, звісно, над цим працюють.

Якщо ви хочете визначити, хто написав уривок — алгоритм машинного навчання чи людина, або принаймні чи текст ретельно редагувала людина, — шукайте серйозні провали пам’яті. Станом на 2019 рік лише деякі системи починали навчатися відстежувати довгострокову інформацію в оповіді, та навіть вони втрачали частину критично важливих подробиць.

Чимало ШІ, що створюють текст, здатні стежити лише за кількома словами одночасно. Ось, наприклад, що написала **рекурентна нейронна мережа (RNN)** після навчання на дев’ятнадцяти тисячах описів людських снів із сайту dreamresearch.net:

> Я встаю й іду коридором до його будинку та бачу птаха в дуже вузькій шухляді, і це група людей у ручних дверях. Удома, наче старший чоловік збирається купити кілька ключів. Він дивиться на свою голову картонним пристроєм, а потім мої ноги припарковані на столі.

Сни відомі своєю непослідовністю: посередині можуть змінитися місце, настрій і навіть персонажі. Проте ці сни нейромережі не зберігають зв’язності довше одного речення, а іноді й значно менше. Персонажів, яких ніхто не представляв, згадують так, наче вони були присутні від початку. Увесь сон забуває, де відбувається. Окремі фрази можуть мати сенс, а ритм слів здається нормальним, якщо не стежити за подіями. Наслідування зовнішніх ознак людської мови за повної відсутності глибшого змісту — характерна риса тексту, створеного нейромережею.

Шістдесят п’ять символів пам’яті й кривавий пудинг

На наступній сторінці наведено ще один приклад — цього разу рецепт, у якому наслідки обмеженої пам’яті помітити ще легше. Його створила та сама рекурентна нейромережа, що згенерувала попередні рецепти в цьому розділі. Як бачите, це модель, яка навчалася на різних стравах, зокрема, очевидно, на рецептах кривавого пудингу — різновиду кров’яної ковбаси. Нейромережа будує рецепт літера за літерою: переглядає вже створені символи й вирішує, який має бути наступним. Але кожна додаткова літера вимагає більше пам’яті, а комп’ютер має обмежені ресурси. Щоб зробити вимоги прийнятними, мережа переглядає лише кілька останніх символів. Для цього алгоритму й мого комп’ютера найбільший доступний обсяг пам’яті становив шістдесят п’ять символів. Щоразу, коли мережа мала додати нову літеру рецепта, вона знала лише попередні шістдесят п’ять.* У тексті добре видно момент, коли вона вичерпала пам’ять і забула, що готує шоколадний десерт: приблизно тоді, коли вирішила додати чорний перець і якусь «рисову сметану».

GPT-2 довше утримує тему

Це обмеження пам’яті починає змінюватися. Дослідники створюють рекурентні нейромережі, які під час передбачення наступних літер тексту враховують і короткострокові, і довгострокові ознаки. Ідея схожа на алгоритми, що спочатку розглядають дрібні елементи зображення — наприклад, краї й текстури, — а потім віддаляються й охоплюють загальну картину. Такі

стратегії називають **згорткою**, або **convolution**. Нейромережа, яка використовує згортки й водночас у сотні разів більша за модель на моєму ноутбуці, може пам’ятати інформацію достатньо довго, щоб не змінювати тему. Наступний рецепт створила нейромережа GPT-2, яку OpenAI навчила на величезній добірці вебсторінок, а я потім додатково навчила на різноманітних рецептах.

The struggle is going to be to remember what it's already written. It sees only 65 characters at a time : from beginning to the first semicolon. Can it at least stick with sweet vs savory?

The format is easy. Start with a title, then do a category, then ingredients, then directions. Predictable. It gets this every time. Predictable things are easy.

```
Chocolate Butterbroth Black Pudding

cheese/eggs

4 oz cocoa; finely ground
1 teaspoon butter
1/2 cup milk
1/4 teaspoon pepper
1/4 cup rice cream, chopped
1 lb cream
1  sesame peel

        ----DATE HOLY----

1 large egg
1  powdered sugar serving barme
1/4 cup butter or margarine, melted

Brown sugar, chocolate; baking powder, beer, lemon juice and salt
in chunk in greased 9 x 2 inch cake.
Chill until golden brown and bubbly.
Place serve garlic half by pieoun on top to make more use bay.
Place in frying pan in preheated oven.
Sprinkle with fresh parsley  for cooking.

Eating dish to hect in pot of the oil, pull over half-and half. Place
in a bowl.   Pat thetings on a strip of calaparo and wanned
cooking in butter by cooking the seasoning. Sprinkle with onions. & Pull
when bubbles and carrot is cooked, about 5
minutes.  On a 15 inch hour a blender or wax paper mix to by dried
roughes to boil (which is discovered.)

Yield: 1 cake
```

This is made from blood. An interesting start.

Okay, I would have gone with "desserts" here. Let's try to stay on track.

Oh good, cocoa! We haven't forgotten about the chocolate yet. It's just barely in the 65-character memory.

Wait, it's already...well, good to make sure, I guess.

Why would we have 1 sesame anything? Clear error here. And peeling it seems tedious.

Sesame brings us dangerously close to savory territory.

Capital letters are a particular challenge because they're treated as completely unrelated to lower case. Neural net has to learn these independently from scratch with very few examples.

Frosting! Date Holy appears to be frosting. It must have lifted this from a cake recipe. We are back in dessert territory.

These weren't in the ingredients list, which has already passed out of memory by now. It got chocolate, but it's guessing.

Uh-oh. Ambiguity. "Until golden brown" could be sweet or savory. "Bubbly" tips the balance and now... Garlic. Game over.

Seems a bit small. And shouldn't this be a pan?

Oh no. It's completely lost track. Not enough info to figure out what's going on. Is this soup? Stir-fry? No good high-probability options. Even its spelling suffers as it flails.

At least it remembered to close its parenthesis. There's probably a neuron just keeping track of parentheses.

Neural net at least knows how to wrap up a recipe. We were making cake, right? Let's say it's cake.

Sometimes it knows it should end the recipe soon because we're adding frosting or serving with something. With this chaos, it must have just guessed. Sometimes these recipes go on for pages, the neural net having no idea how long it has been.

Торт із шматками, торти, пустелі
8 склянок борошна
4 фунти масла кімнатної температури
2 ¼ склянки кукурудзяного сиропу, розділити
2 яйця, пюровані й охолоджені
1 чайна ложка винного каменю
½ склянки M&M’s
8 унцій шматкуватих білих
1 шоколад, просіяний

> Збийте 2 ¼ склянки борошна на середній швидкості до загустіння.
> Злегка змастіть і присипте борошном два змащені й вистелені вощеним папером коробкові інгредієнти; з’єднайте борошно, сироп і яйця. Додайте винний камінь. Вилийте у форму для хліба об’ємом один галон. Випікайте при 230 °C 35 хвилин. Тим часом у великій мисці з’єднайте сироп, білі й шоколад; перемішайте який доки все ретельно не змішається. Остудіть форму. Вилийте 2 столові ложки шоколадної суміші на весь торт. Тримайте в холодильнику до подавання.
> Вихід: 20 порцій

Завдяки поліпшеній згортками пам’яті GPT-2 пам’ятає про більшість інгредієнтів і навіть не забуває, що готує торт. Указівки все одно дещо неймовірні: звичайне борошно не загусне, хоч би як довго його збивати, а суміш борошна, сиропу та яєць навряд чи стане тортом навіть після додавання винного каменю. Та порівняно із шоколадно-масляним бульйоном із кривавим пудингом це разючий поступ.

> Ось іще один приклад GPT-2 — цього разу спроба написати фанфік про Гаррі Поттера. Алгоритм зміг відстежувати, які персонажі перебувають у сцені, та навіть запам’ятав повторюваний мотив: на голові Снейпа вже була змія.

> Снейп: Я розумію.
> [З’являється змія, Снейп кладе її собі на голову, і здається, що говорить саме вона. Вона каже: «Я тебе пробачаю».]
> ГАРРІ: Ти не можеш повернутися, якщо не пробачиш.
> Снейп: [зітхає] Герміоно.
> ГАРРІ: Гаразд, послухай.
> Снейп: Я хочу перепросити перед тобою за те, що розсердився й засмутився через це.
> ГАРРІ: Це не твоя провина.
> ГАРРІ: Я не це хотів сказати.
> [З’являється ще одна змія й каже: «І я тебе пробачаю».]
> ГЕРМІОНА: І я тебе пробачаю.
> Снейп: Так.

Як групування одиниць допомагає пам’яті

Ще один спосіб упоратися з обмеженнями пам’яті — об’єднати базові одиниці в групи, щоб нейромережа могла зберігати зв’язність, пам’ятаючи менше окремих елементів. Замість шістдесяти п’яти літер вона могла б утримувати шістдесят п’ять цілих слів або навіть шістдесят п’ять елементів сюжету. Якби я обмежила свою мережу спеціально складеним переліком обов’язкових інгредієнтів і допустимих діапазонів — саме так зробила команда Google, яка намагалася створити нове безглютенове печиво з шоколадними шматочками, — алгоритм щоразу видавав би припустимий рецепт.[17] На жаль, результат

Google, хоч і був значно більше схожий на печиво, ніж будь-що створене моїм алгоритмом, за повідомленнями все одно виявився жахливим.[18]

ЧИ Є ПРОСТІШИЙ СПОСІБ РОЗВ’ЯЗАТИ ЦЮ ПРОБЛЕМУ?

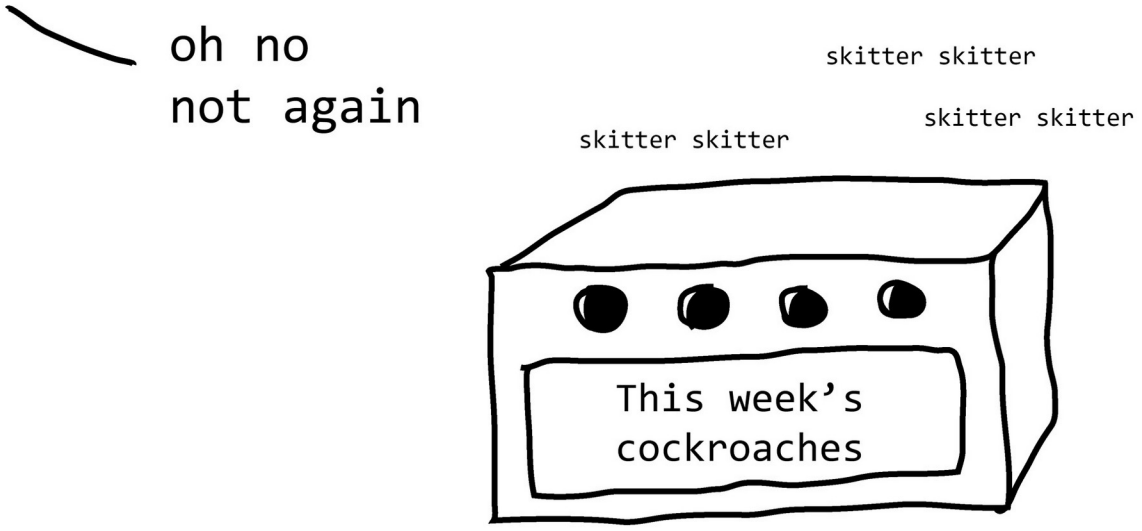
Коли ШІ справді є найпростішим рішенням

Так ми підходимо до одного з останніх чинників, що визначають придатність проблеми для ШІ. Щоправда, він не визначає, чи спробують люди все одно застосувати ШІ: чи справді це найпростіший спосіб розв’язання?

До появи великих моделей і величезних наборів даних у деяких проблемах майже не вдавалося досягти поступу. ШІ докорінно змінив розпізнавання зображень і машинний переклад, завдяки чому розумні мітки на фотографіях і Google Translate стали повсюдними. Людям важко записати загальні правила для таких завдань, натомість ШІ може проаналізувати багато інформації й створити власні. Або він може переглянути сто характеристик клієнтів мобільного оператора, які перейшли до конкурента, і визначити, хто, найімовірніше, піде наступним. Можливо, непостійні клієнти молоді, живуть у районах із гіршим за середнє покриттям і користуються послугами менш ніж пів року.

Коли замість ШІ достатньо здорового глузду

Небезпечно застосовувати складне рішення на основі ШІ там, де краще спрацював би звичайний здоровий глузд. Можливо, йдуть саме клієнти, підписані на щотижневу доставку тарганів. Адже цей тариф *жахливий*.



ДОЗВОЛИТИ ШІ КЕРМУВАТИ?

Чому безпілотне керування так приваблює

А як щодо безпілотних автомобілів? Це завдання привабливе для ШІ з багатьох причин. Ми, звісно, хотіли б автоматизувати водіння: чимало людей вважають його нудним, а дехто часом узагалі не може керувати. Компетентний ШІ-водій мав би блискавичні рефлекси, ніколи не виляв би в межах смуги, не виїжджав за неї й не поводився агресивно. Насправді безпілотні автомобілі часом бувають навіть *надто* обережними: їм важко влитися в щільний потік у годину пік або повернути ліворуч на жвавій дорозі.[19] Зате ШІ ніколи не втомився б і міг би безкінечно тримати кермо, поки люди сплять або розважаються.

Можна накопичити багато прикладів, якщо ми готові платити водіям-людям, щоб ті проїхали мільйони кілометрів. Також легко побудувати віртуальні симуляції, де ШІ випробовуватиме й удосконалюватиме стратегії в прискореному часі.

Вимоги водіння до пам’яті теж відносно скромні. Поточний поворот керма й швидкість не залежать від того, що сталося п’ять хвилин тому. За планування майбутнього відповідає навігація. Небезпеки на дорозі — пішоходи й тварини — з’являються та зникають за лічені секунди.

Нарешті, керувати автомобілем настільки складно, що інших хороших рішень ми не маємо. Саме ШІ просунув нас найдалі.

Рідкісні втручання людини, яких не вдається позбутися

Однак досі незрозуміло, чи водіння достатньо вузьке для сучасного ШІ, чи воно потребує чогось ближчого до загального штучного інтелекту людського рівня. Автомобілі під керуванням ШІ вже довели, що здатні самостійно долати мільйони кілометрів, а деякі компанії повідомляли: під час випробувань людина втручалася лише раз на кілька тисяч кілометрів. Проте саме цю рідкісну потребу у втручанні й виявилося надзвичайно важко усунути остаточно.

Людям доводилося рятувати ШІ безпілотних автомобілів у найрізноманітніших ситуаціях. Зазвичай компанії розкривають лише кількість таких «відключень», як цього вимагає закон у деяких місцях, але не пояснюють причин. Можливо, частково тому, що вони бувають лячно буденними. У дослідженні 2015 року[20] перелічено кілька випадків. Серед іншого автомобілі:

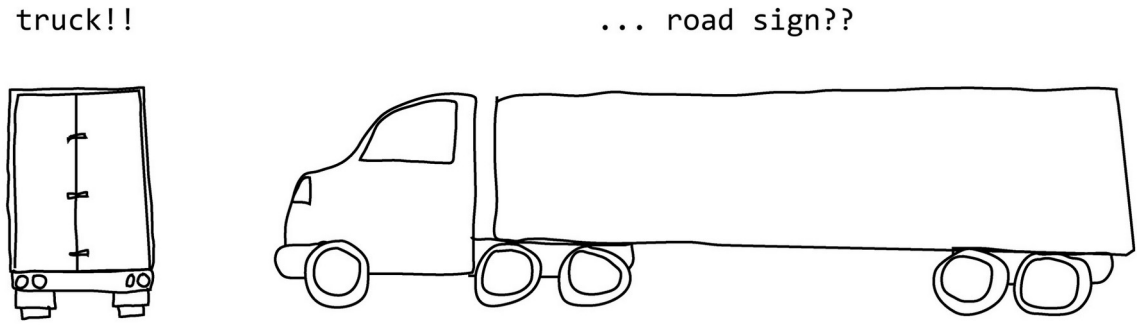
- сприймали навислі гілки як перешкоду;

- плуталися, у якій смузі рухається інша машина;
- вирішували, що на перехресті забагато пішоходів;
- не бачили автомобіля, який виїжджав із паркінгу;
- не помічали автомобіля, що виїхав просто перед ними.

Фатальні помилки розпізнавання перешкод

Смертельна аварія в березні 2018 року стала наслідком схожої ситуації: ШІ безпілотного автомобіля не міг правильно розпізнати пішоходку. Спочатку класифікував її як невідомий об’єкт, потім як велосипед і лише за 1,3 секунди до моменту, коли треба було гальмувати, — як пішоходку. Ситуацію погіршило те, що систему аварійного гальмування автомобіля вимкнули, віддавши перевагу сповіщенню резервного водія, хоча сама система не була спроєктована так, щоб справді його попереджати. До того ж резервний водій провів багато-багато годин у машині без жодної потреби втручатися — обставина, за якої переважна більшість людей утратила б пильність.[21] Смертельна аварія 2016 року також сталася через помилку розпізнавання перешкоди: безпілотний автомобіль не визначив вантажівку з платформою як небезпеку. Докладніше — у вставці нижче.

> 2016 року сталася смертельна аварія, коли водій використав автопілот Tesla на міських вулицях, хоча систему призначали для шосе. Перед автомобілем проїхала вантажівка, але ШІ автопілота не загальмував: він не зареєстрував її як перешкоду, якої потрібно уникнути. За аналізом Mobileye — компанії, що розробила систему запобігання зіткненням, — її навчали лише уникати наїздів ззаду, оскільки вона була призначена для руху шосе. Тобто ШІ вчився розпізнавати вантажівки ззаду, а не збоку. Tesla повідомила, що система побачила вантажівку, але прийняла її за дорожній знак над смугою й вирішила, що гальмувати не потрібно.[22]



Кенгуру, ему, лава й нескінченність рідкісних ситуацій

І це ще без урахування незвичайних подій. Коли Volkswagen уперше випробував свій ШІ в Австралії, з’ясувалося, що його збивають із пантелику

кенгуру. Очевидно, раніше система не зустрічала нічого, що пересувається стрибками.[23]

На дорозі може відбутися неймовірне розмаїття подій: парад, утеча ему, обірвані лінії електропередач, лава, аварійні знаки з незвичними вказівками, повінь із меляси або провалля. Неминуче станеться щось, чого ШІ ніколи не бачив під час навчання. Дуже важко створити систему, здатну впоратися з цілковитою несподіванкою: інтуїтивно зрозуміти, що ему-втікач, найімовірніше, хаотично бігатиме, а провалля залишиться на місці; усвідомити, що хоча лава тече й збирається в калюжі приблизно як вода, це не означає, що через неї можна проїхати.

Контрольовані маршрути наближають нас до громадського транспорту

Автовиробники пристосовують свої стратегії до неминучості буденних збоїв і надзвичайних дивовиж на дорозі. Вони розглядають обмеження безпілотних автомобілів закритими контрольованими маршрутами — хоча це не обов’язково розв’яже проблему ему, адже вони хитрі, — або рух безпілотних вантажівок колоною за провідним водієм-людиною. Інакше кажучи, компроміси ведуть до рішень, дуже схожих на масовий громадський транспорт.

Autonomy levels of self-driving cars

```
0. No Automation           Fixed-speed cruise control, at most.
                           A Model T Ford qualifies.
                           You're driving, end of story.

1. Driver Assistance       Adaptive cruise control or lane-keeping.
                           Most modern cars have this.
                           Some part of you is driving.

2. Partial Automation      2 or more things from level 1 work together.
                           Car can maintain distance AND follow the road.
                           Driver still must be ready to take over.

3. Conditional Automation  Car can drive by itself in some conditions.
                           Cars with traffic jam mode; highway mode.
                           Driver rarely needed but must be READY.

4. High Automation         Car doesn't need a driver on a controlled route.
                           Sometimes, driver can go in back and nap.
                           On other routes, still needs a driver.

5. Full Automation         Car never needs a driver.
                           Car might not even have wheels and pedals.
                           Go back to sleep. Car has it all under control.
```

Проблема автоматизації третього рівня

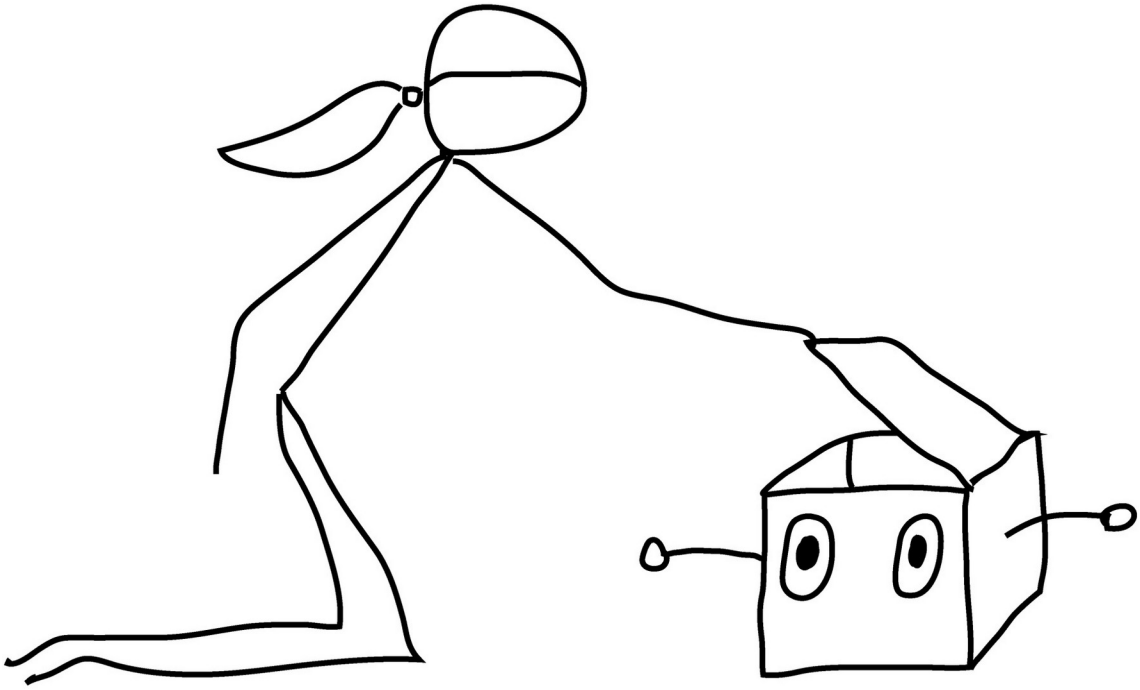
На час написання книги, коли ШІ розгублювався, він відключався — тобто раптово повертав керування людині за кермом. Третій рівень, умовна автоматизація, був найвищим комерційно доступним рівнем автономності. Наприклад, у режимі автопілота Tesla автомобіль міг годинами їхати самостійно, але людину могли будь-якої миті попросити перебрати керування. Проблема такого рівня в тому, що водій має сидіти за кермом і стежити за дорогою, а не прикрашати печиво на задньому сидінні. Люди дуже-дуже погано зберігають пильність після кількох нудних годин пасивного спостереження. Допомога людини часто непогано долає розрив між наявною й потрібною ефективністю ШІ, але люди дуже погано рятують безпілотні автомобілі.

Безпілотні автомобілі потребують компромісів або гнучкішого ШІ

Отже, створення безпілотних автомобілів — водночас привабливе й надзвичайно складне завдання для ШІ. Щоб вони стали масовими, можливо, доведеться піти на компроміси — наприклад, створити контрольовані маршрути й обмежитися четвертим рівнем автоматизації — або ж знадобиться значно гнучкіший ШІ, ніж ми маємо нині.

У наступному розділі ми розглянемо типи ШІ, що лежать в основі безпілотних автомобілів та інших систем: змодельовані за принципами роботи мозку, еволюції й навіть гри «вірю — не вірю».

РОЗДІЛ 3. Як він насправді навчається?



Як випадкові літери стають жартами

Пам’ятайте, що в цій книжці я вживаю термін *ШІ* у значенні «програми машинного навчання». (Зручна таблиця тут показує, що я вважаю і не вважаю ШІ. Вибачте, людино в костюмі робота.) Як я пояснювала в розділі 1, програма машинного навчання розв’язує задачу методом спроб і помилок. Але як саме це відбувається? Як програма переходить від мішанини випадкових літер до впізнаваних жартів «тук-тук» — і все це без людини, яка пояснила б їй, як працюють слова або що взагалі таке жарт?

Існує чимало різних методів машинного навчання, багатьом із яких уже десятки років: вони з’явилися задовго до того, як люди почали називати їх ШІ. Сьогодні ці технології поєднують, переробляють і посилюють швидшими обчисленнями та більшими наборами даних. У цьому розділі ми розглянемо кілька найпоширеніших типів і зазирнемо їм під капот, щоб побачити, як вони навчаються.

НЕЙРОННІ МЕРЕЖІ

Сьогодні, говорячи про ШІ або **глибоке навчання**, люди часто мають на увазі **штучні нейронні мережі (ШНМ)**. (Їх також називали **кібернетикою** або **конекціонізмом**.)

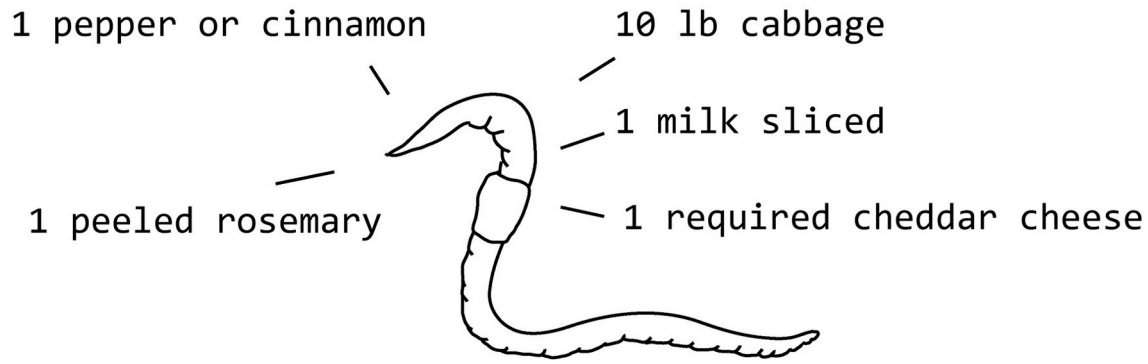
Штучні нейронні мережі можна будувати багатьма способами — кожен для свого застосування. Одні спеціалізуються на розпізнаванні зображень, інші — на обробці мови, створенні музики, підвищенні продуктивності тарганячої ферми чи написанні незрозумілих жартів. Але всі вони дуже приблизно наслідують роботу мозку. Тому їх і називають штучними нейронними мережами: їхні родичі, **біологічні нейронні мережі**, — це первісні й набагато складніші моделі. Ба більше, коли в 1950-х роках програмісти створювали перші штучні нейромережі, вони прагнули перевірити теорії про роботу мозку.

Інакше кажучи, штучні нейронні мережі — це мозки-імітації.

Мозок потужністю в одного черв’яка

Вони складаються з безлічі простих фрагментів програмного забезпечення, кожен із яких виконує дуже прості математичні операції. Ці фрагменти зазвичай називають **клітинами** або **нейронами** — за аналогією з нейронами нашого мозку. Сила нейронної мережі полягає в тому, як ці клітини сполучені.

Порівняно зі справжнім людським мозком штучні нейронні мережі не надто потужні. Ті, якими я створювала багато текстів у цій книжці, мають стільки ж нейронів, скільки… черв’як.



На відміну від людини, нейромережа принаймні здатна присвятити весь свій мозок потужністю в одного черв’яка поточному завданню (якщо я випадково не відверну її сторонніми даними). Але як розв’язувати задачі за допомогою купи взаємопов’язаних клітин?

Чому людська кількість нейронів ще не дасть людського інтелекту

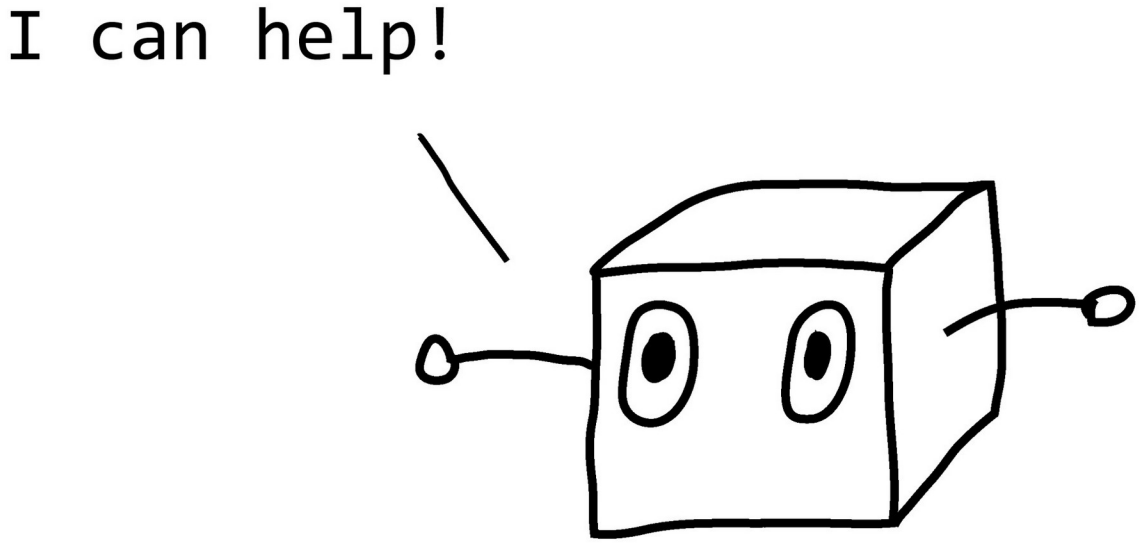
> Найпотужніші нейронні мережі, навчання яких триває місяцями й коштує десятки тисяч доларів обчислювального часу, мають набагато більше нейронів, ніж нейромережа на моєму ноутбуці; деякі навіть перевищують кількість нейронів в однієї медоносної бджоли. Простеживши, як із часом зростали найбільші нейронні мережі світу, один провідний дослідник у 2016 році оцінив, що приблизно до 2050 року штучні мережі можуть наблизитися за кількістю нейронів до людського мозку.[1] Чи означає це, що тоді ШІ наблизиться до людського інтелекту? Найімовірніше, навіть близько ні. Кожен нейрон людського мозку набагато складніший за нейрон штучної мережі — настільки, що окремий людський нейрон радше нагадує цілу багатошарову нейромережу. Отже, людський мозок — не просто нейронна мережа з вісімдесяти шести мільярдів нейронів, а мережа з вісімдесяти шести мільярдів нейронних мереж. І наш мозок має набагато більше складностей, ніж ШНМ, зокрема чимало таких, яких ми досі не розуміємо.

МАГІЧНА СЕНДВІЧНА ДІРА

Один чудовий сендвіч на тисячу жахливих

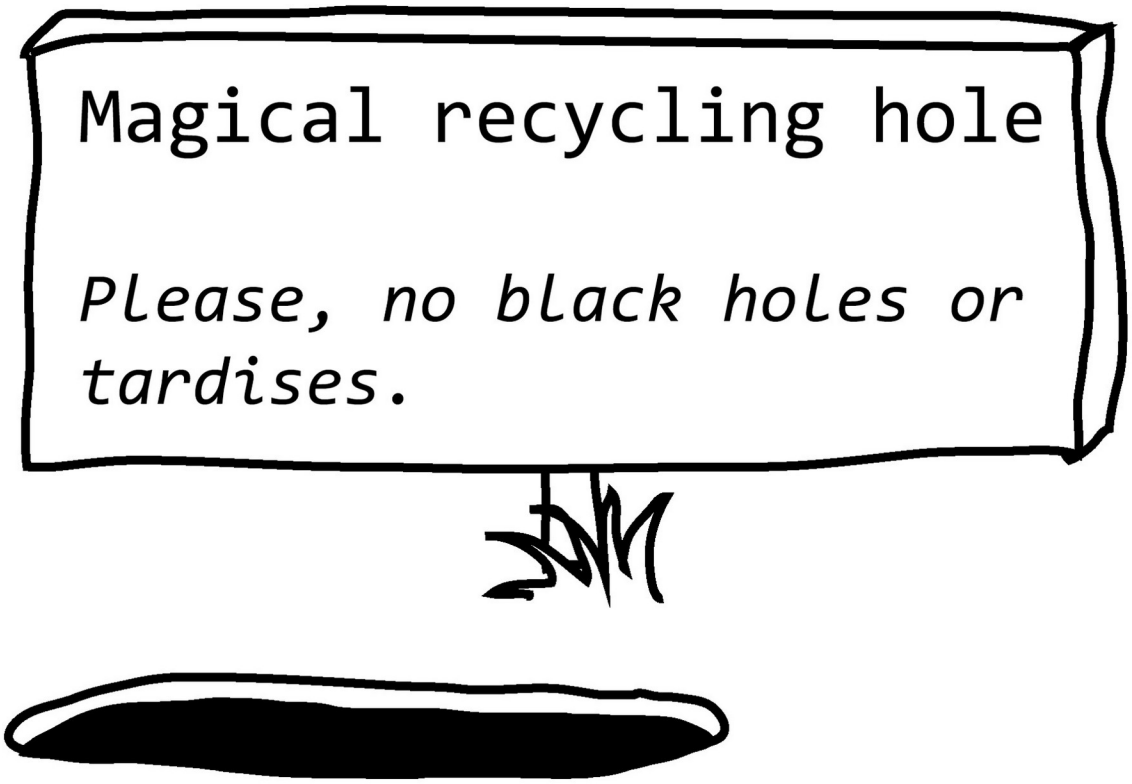
Припустімо суто гіпотетично, що ми знайшли в землі магічну діру, яка кожні кілька секунд видає випадковий сендвіч. (Гаразд, це *дуже* гіпотетично.) Проблема в тому, що сендвічі надзвичайно випадкові. Серед інгредієнтів трапляються варення, кубики льоду й старі шкарпетки. Щоб знайти добрі сендвічі, нам доведеться цілий день сидіти перед дірою та сортувати їх.

Але це швидко набридне. Добрим виявляється лише один сендвіч із тисячі. Зате вже якщо він добрий, то *дуже-дуже*. Спробуймо автоматизувати роботу.



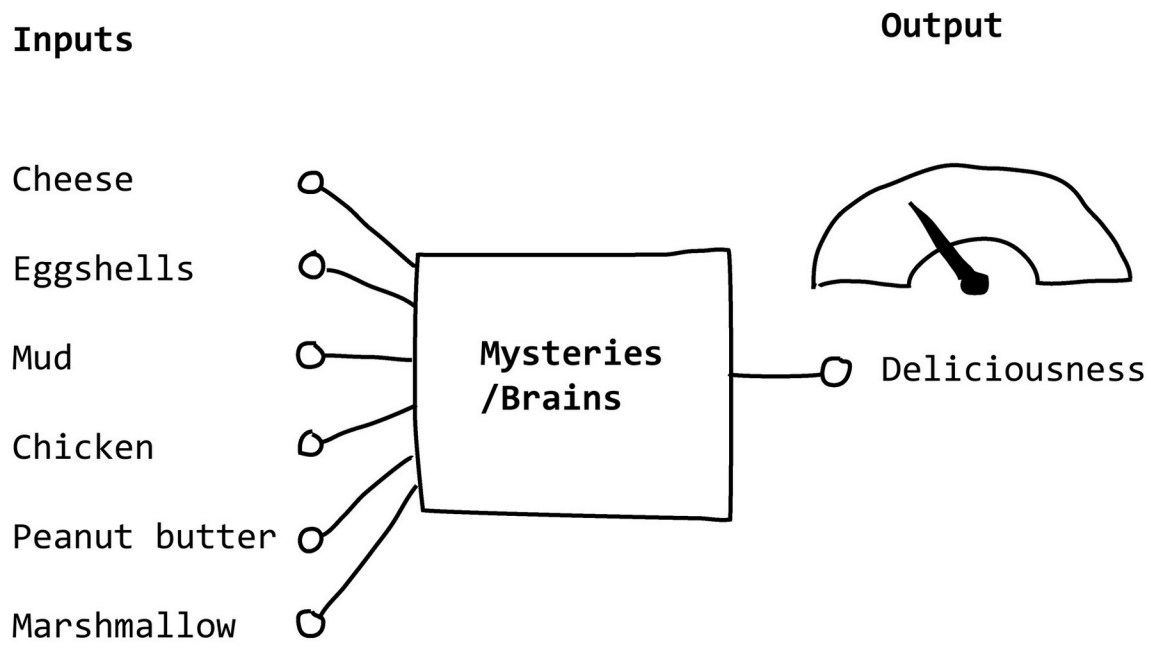
Щоб заощадити час і сили, ми хочемо побудувати нейронну мережу, яка розглядатиме кожен сендвіч і вирішуватиме, чи він добрий. Поки що не зважаймо на те, як змусити нейромережу розпізнавати його інгредієнти: це дуже складна задача. І не зважаймо на те, як вона братиме кожен сендвіч. Це теж неймовірно складно — треба не лише розпізнати рух сендвіча, що вилітає з діри, а й наказати роботизованій руці схопити тонкий сендвіч із папером і моторною оливою або товстий — із кулею для боулінгу та гірчицею. Тож припустімо, що нейромережа знає склад кожного сендвіча, а проблему їх фізичного переміщення ми вже розв’язали. Їй залишається тільки вирішити:

зберегти цей сендвіч для людини чи викинути в жолоб для перероблення. (Механізм жолоба теж ігноруємо — нехай це буде ще одна магічна діра.)

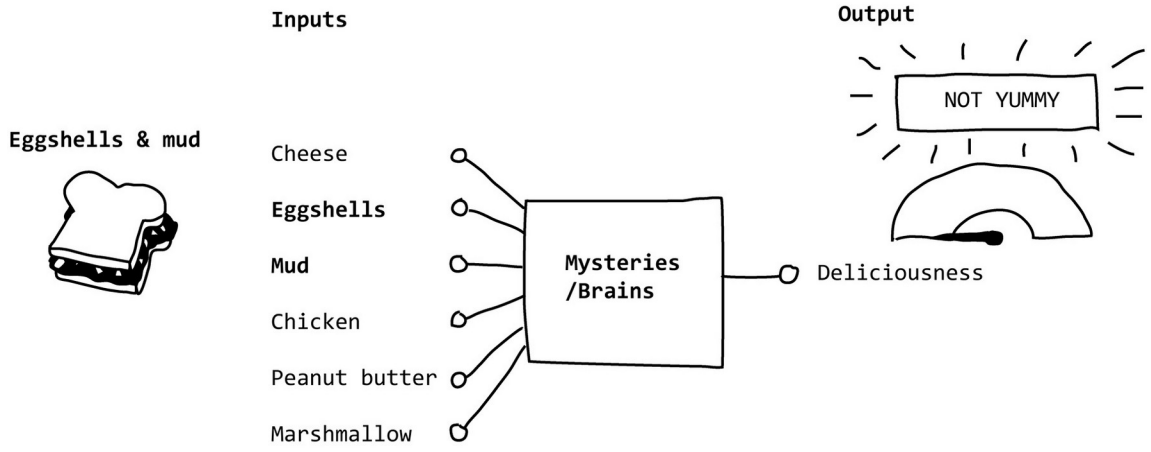


Чорна скринька з інгредієнтами на вході й смакотою на виході

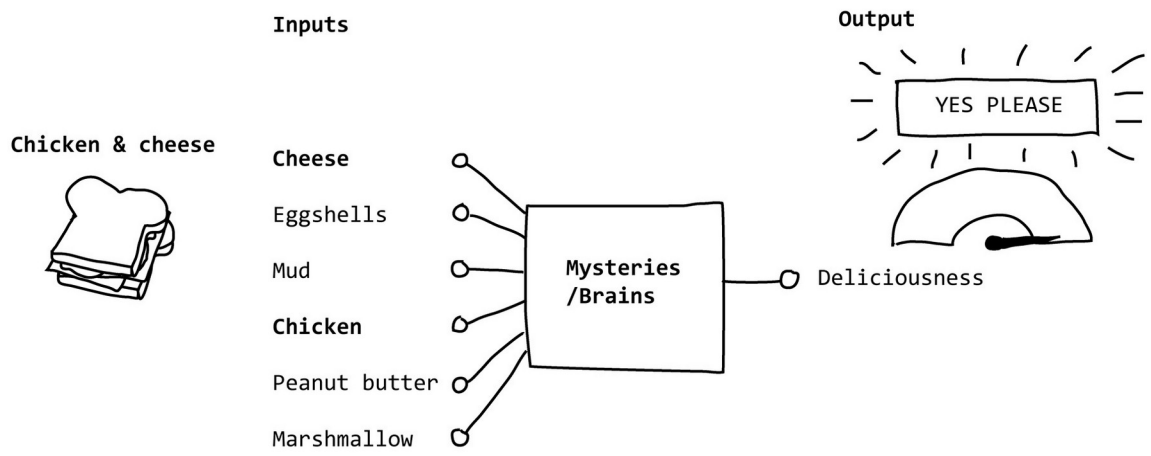
Так ми звели завдання до простого й вузького — а в розділі 2 з’ясували, що саме такі завдання добре підходять для автоматизації алгоритмом машинного навчання. Маємо низку вхідних даних (назви інгредієнтів) і хочемо побудувати алгоритм, який на їх основі обчислить єдиний вихід — число, що показує, чи добрий сендвіч. Простий малюнок нашого алгоритму як «чорної скриньки» має такий вигляд:



Ми хочемо, щоб значення «смакоти» на виході змінювалося залежно від поєднання інгредієнтів. Тож якщо в сендвічі є яєчна шкаралупа та багнюка, наша чорна скринька має зробити ось що:



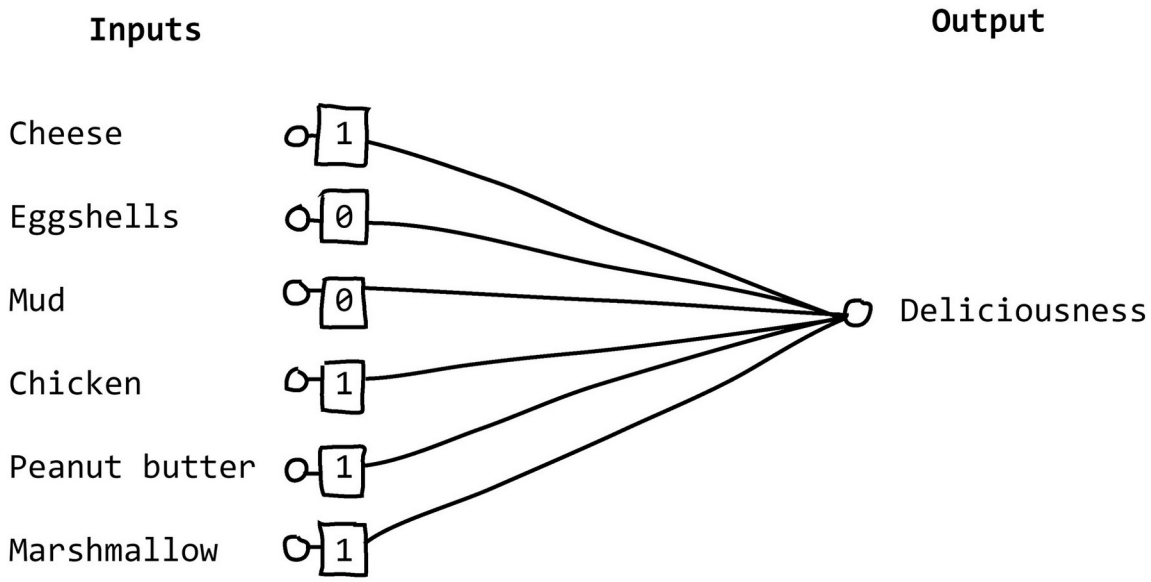
А якщо в ньому курятина й сир — ось це:



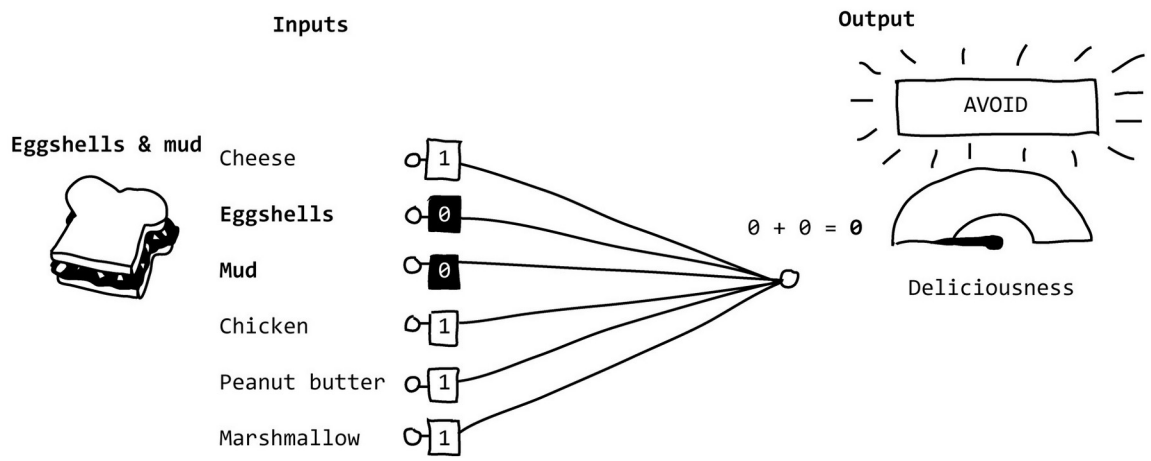
Погляньмо, як усе з’єднано всередині чорної скриньки.

Проста мережа додає ваги інгредієнтів

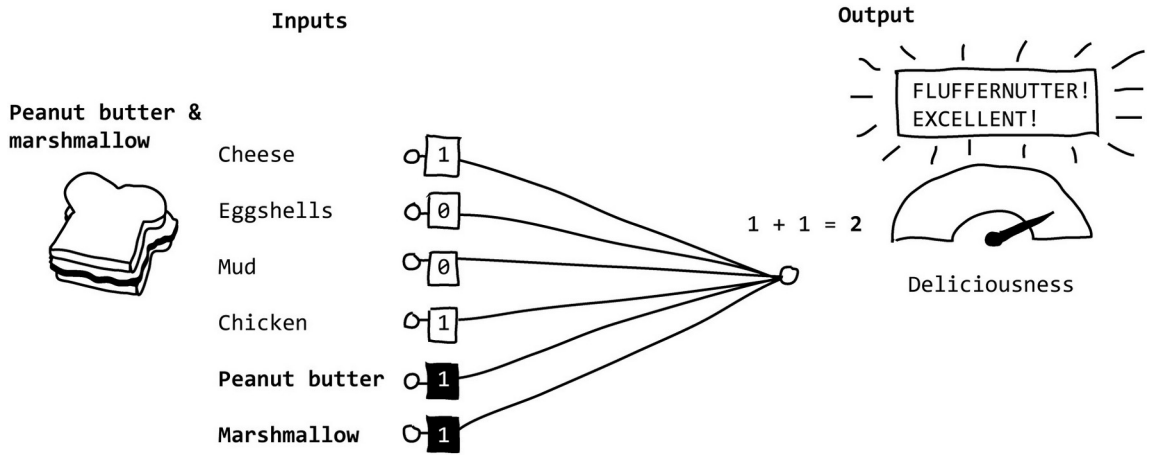
Спочатку зробімо все просто. Під’єднаємо всі входи (усі інгредієнти) до єдиного виходу. Щоб отримати оцінку смакоти, складемо внески всіх інгредієнтів. Очевидно, вони не мають бути однаковими: сир зробить сендвіч смачнішим, а багнюка — менш смачним. Тому кожен інгредієнт дістає свою вагу. Добрим призначаємо вагу 1, а тим, яких хочемо уникнути, — 0. Наша нейронна мережа має такий вигляд:



Перевірмо її на кількох сендвічах. Припустімо, у сендвічі є багнюка та яєчна шкаралупа. Обидва інгредієнти дають 0, тож оцінка смакоти становить 0 + 0 = 0.



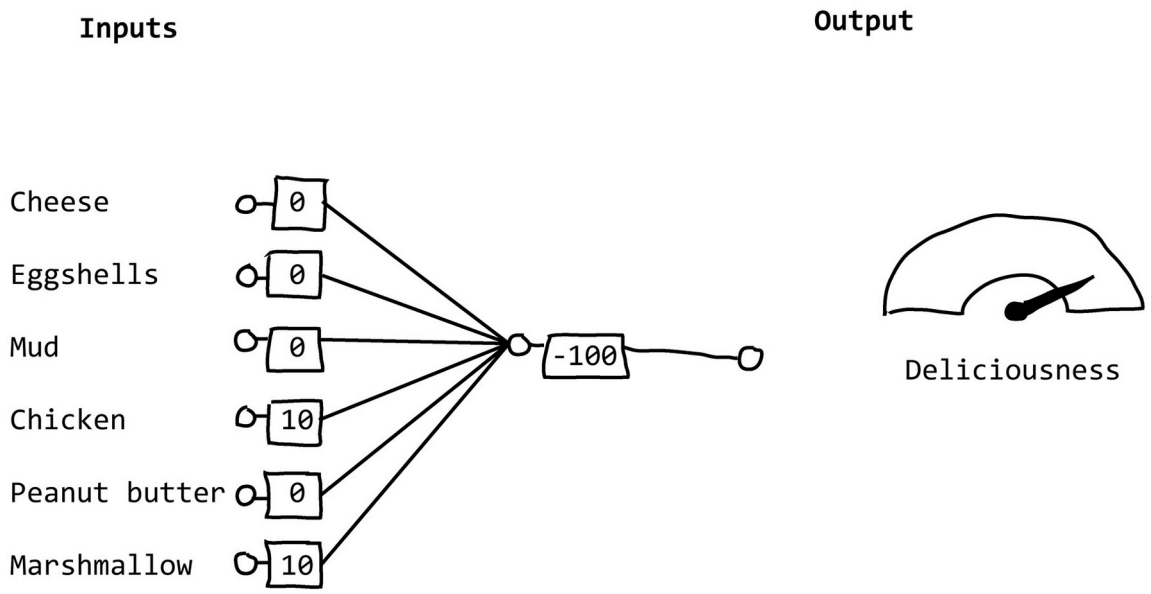
А сендвіч з арахісовою пастою та маршмелоу отримає 1 + 1 = 2. (Вітаю! Вас благословили новоанглійським делікатесом — флафернатером.)



Помилка великого сендвіча: смакота може «скасувати» мульчу

З такою конфігурацією нейромережі ми успішно уникаємо сендвічів, що містять тільки шкаралупу, багнюку та інші неїстівні речі. Але ця проста одношарова мережа недостатньо витончена, щоб зрозуміти: деякі інгредієнти, смачні окремо, у певних поєднаннях уже не смакують. Курятину з маршмелоу вона оцінить як смакоту, рівну флафернатеру. Вона також вразлива до того, що ми назвемо **помилкою великого сендвіча**: сендвіч із мульчею може все одно отримати оцінку «смачний», якщо містить достатньо добрих інгредієнтів, здатних урівноважити мульчу.

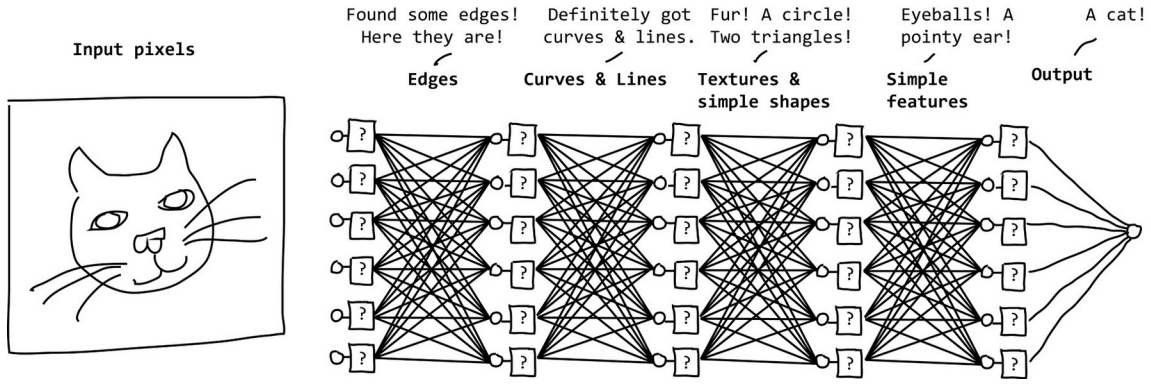
Щоб поліпшити нейронну мережу, нам знадобиться ще один шар клітин.



Прихований шар навчається поєднувати ознаки

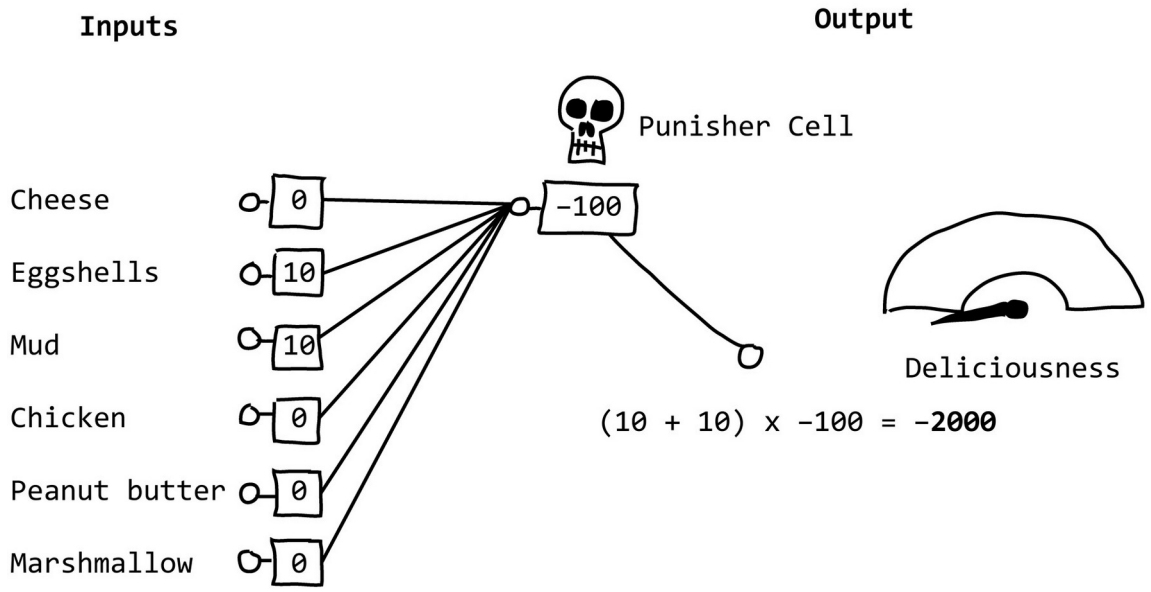
Ось який вигляд тепер має наша мережа. Кожен інгредієнт з’єднаний із новим шаром клітин, а кожна клітина — з виходом. Новий шар називають **прихованим**, бо користувач бачить лише входи й виходи. Як і раніше, кожен зв’язок має власну вагу, тож по-різному впливає на остаточну оцінку смакоти. Це ще не глибоке навчання (для нього потрібно більше шарів), але ми наближаємося.

> **ГЛИБОКЕ НАВЧАННЯ**
>
> Додаючи до нейромережі приховані шари, ми отримуємо складніший алгоритм, здатний оцінювати сендвіч як щось більше, ніж суму інгредієнтів. У цьому розділі ми додали тільки один прихований шар, але реальні нейромережі часто мають кілька. Кожен новий шар дає новий спосіб поєднати висновки попереднього — сподіваємося, на дедалі вищих рівнях складності. Такий підхід — багато прихованих шарів для великої складності — називають **глибоким навчанням**.



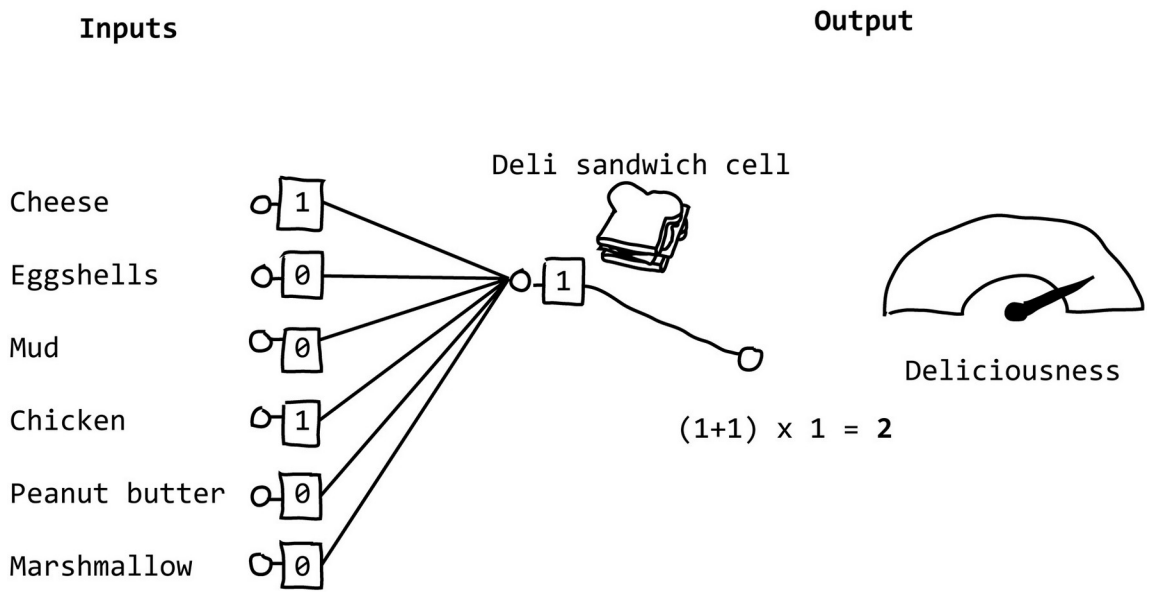
Клітина-каральник перемагає помилку великого сендвіча

Тепер ми нарешті можемо уникати поганих інгредієнтів, під’єднавши їх до клітини, яку назвемо каральником. Призначимо їй величезну від’ємну вагу (скажімо, –100), а все погане з’єднаємо з нею вагою 10. Нехай перша клітина стане каральником, до якого під’єднані багнюка та шкаралупа. Ось як це виглядає:



Тепер сендвіч, що містить яєчну шкаралупу або багнюку, найімовірніше, буде забраковано незалежно від того, що відбувається в інших клітинах. Клітина-каральник дає нам змогу перемогти помилку великого сендвіча.

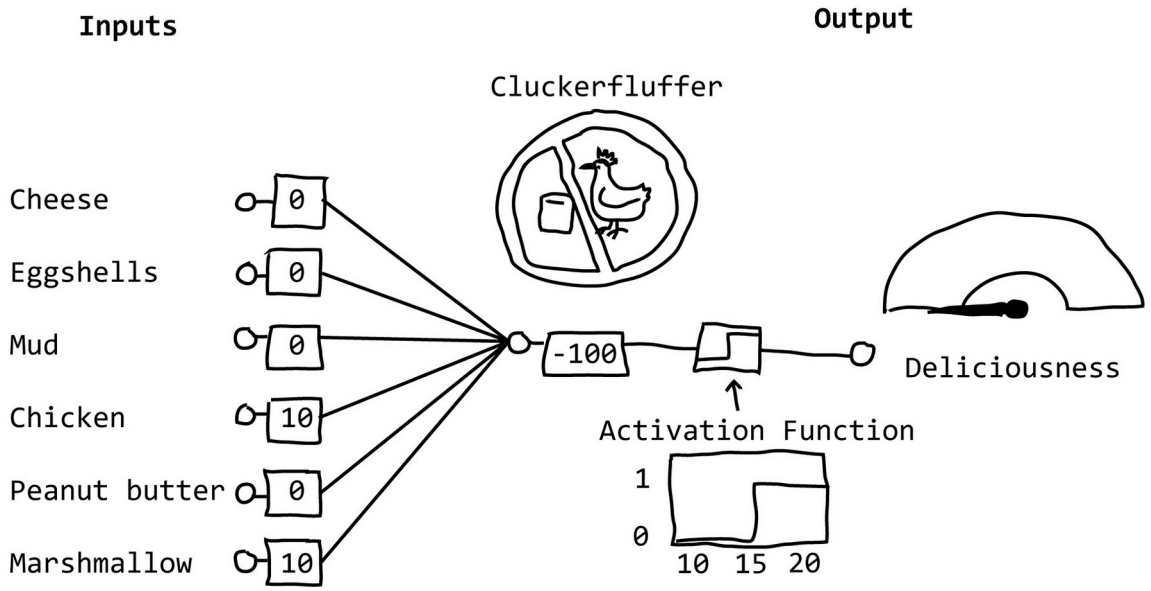
З рештою клітин можна зробити інше — наприклад, нарешті навчити мережу розуміти вдалі поєднання інгредієнтів. Скористаймося другою клітиною, щоб розпізнавати сендвічі на кшталт «курятина із сиром». Назвемо її клітиною делікатесного сендвіча. З’єднаємо з нею курятину й сир вагами 1 (а також шинку, індичку та майонез), а все інше — вагами 0. Саму клітину під’єднаємо до виходу помірною вагою 1. Делікатесний сендвіч — це добре, та якщо ми надто захопимося й дамо цій клітині дуже високу вагу, то ризикуємо послабити каральника. Погляньмо, що вона робить.



Сендвіч із курятиною та сиром змусить клітину радісно додати до кінцевого результату 1 + 1 = 2. Але якщо додати до нього маршмелоу, оцінка не постраждає, хоча об’єктивно сендвіч стане помітно менш смачним. Щоб це виправити, потрібні інші клітини, які шукатимуть і каратимуть конкретні несумісності.

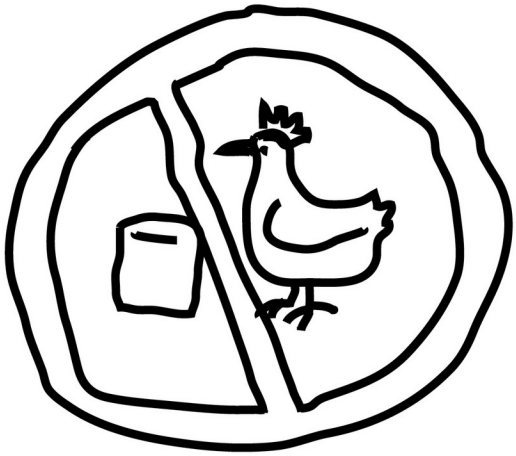
Клакерфлафер карає саме курятину з маршмелоу

Наприклад, клітина 3 може шукати поєднання курятини з маршмелоу (назвімо її «клакерфлафер») і суворо карати будь-який сендвіч із ними обома. Під’єднано її буде так:

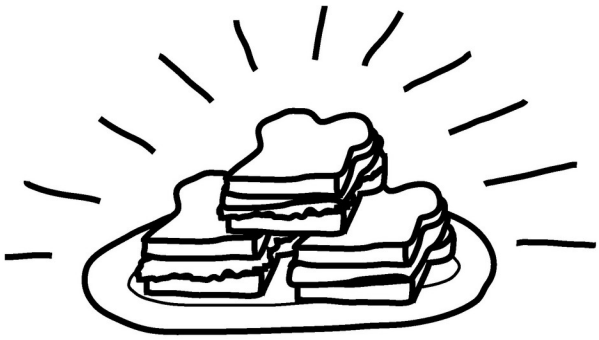


Клітина 3 повертає нищівне (10 + 10) × –100 = –2000 будь-якому сендвічу, що насмілиться поєднати курятину з маршмелоу. Вона діє як вузькоспеціалізований каральник, створений саме для покарання цієї пари. Зауважте, я показала тут додаткову частину клітини клакерфлафера — **функцію активації**, бо без неї клітина карала б *кожен* сендвіч, у якому є курятина *або* маршмелоу. Поріг 15 не дає їй увімкнутися за наявності лише курятини (10 балів) чи лише маршмелоу (10 балів): результат буде нейтральним нулем. Але якщо присутні *обидва* (10 + 10 = 20 балів), поріг 15 перевищено й клітина вмикається. *Бум!* Активована клітина карає будь-яке поєднання інгредієнтів, що перевищує її поріг.

Cluckerfluffer



Коли всі клітини з’єднані в настільки ж хитромудрих конфігураціях, ми отримуємо нейромережу, здатну відібрати найкращі сендвічі, які пропонує магічна діра.

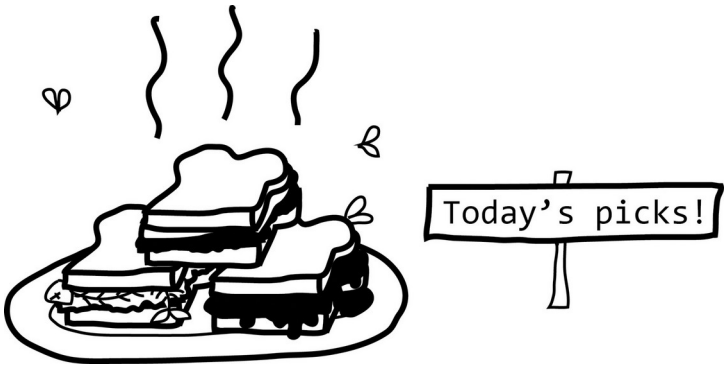


ПРОЦЕС НАВЧАННЯ

Тепер ми знаємо, який вигляд може мати добре налаштована нейромережа для відбору сендвічів. Але сенс машинного навчання в тому, що нам не потрібно налаштовувати її вручну. Вона має *сама* сконфігурувати себе так, щоб чудово вибирати сендвічі. Як відбувається це навчання?

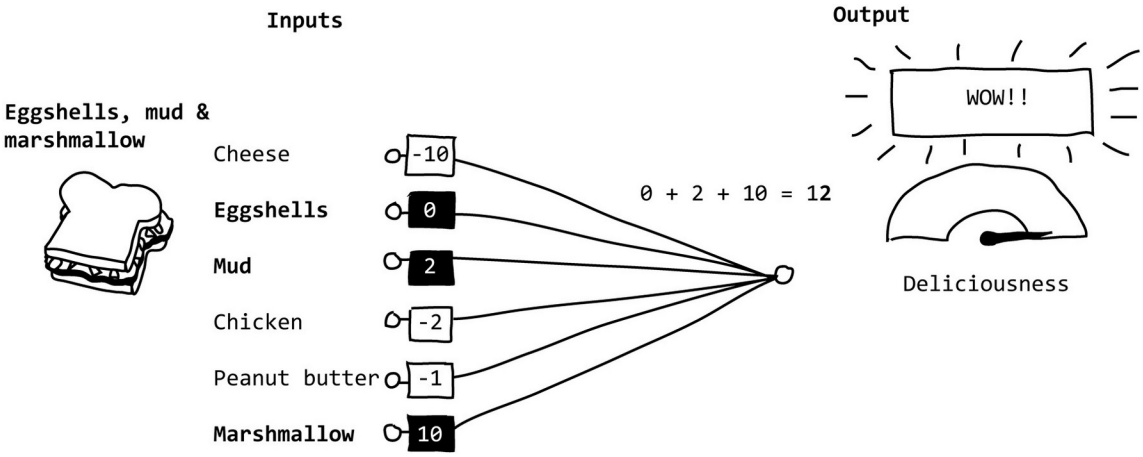
Навчання починається з випадкових і дуже поганих ваг

Повернімося до простої двошарової нейромережі. На початку навчання вона стартує з нуля, з випадковою вагою для кожного інгредієнта. Найімовірніше, оцінює сендвічі вона жахливо.



Потрібно навчити її на реальних даних — прикладах правильного оцінювання сендвічів, наданих справжніми людьми. Оцінюючи кожен сендвіч, нейромережа має порівнювати свій бал з оцінкою колегії послужливих дегустаторів. Примітка: ніколи не зголошуйтеся випробовувати ранні версії алгоритму машинного навчання.

Для цього прикладу знову візьмемо найпростішу нейронну мережу. Оскільки ми намагаємося навчити її з нуля, то ігноруємо всі попередні знання про те, якими мають бути ваги, і починаємо з випадкових. Ось вони:

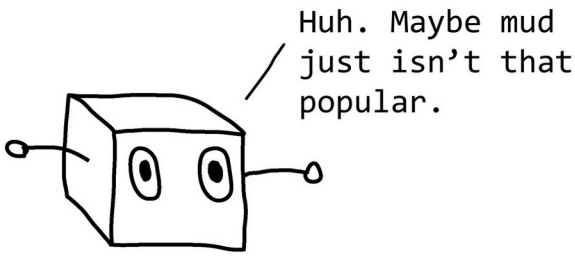


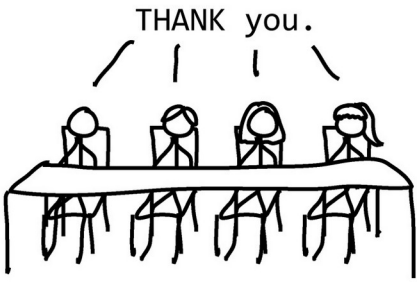
Вона *ненавидить* сир. *Обожнює* маршмелоу. Досить прихильна до багнюки. А яєчна шкаралупа їй байдужа.

Нейромережа розглядає перший сендвіч із магічної діри й, спираючись на своє (жахливе) судження, ставить оцінку. Це сендвіч із маршмелоу, шкаралупою та багнюкою, тож він отримує 10 + 0 + 2 = 12. Ого! Справді неймовірний результат!

Вона подає сендвіч колегії людей-суддів. Сувора реальність: популярності він не здобуває.

Тепер нейромережа має нагоду поліпшитися: вона дивиться, що сталося б, якби її ваги трохи відрізнялися. За одним сендвічем вона не розуміє, у чому проблема. Чи надто захопилася маршмелоу? Може, шкаралупа не нейтральна, а бодай трішечки погана? Вона не знає. Та якщо розглянути партію з десяти сендвічів, оцінки мережі й оцінки людей, можна виявити, що загальне зниження ваги багнюки — і, відповідно, оцінки кожного сендвіча з нею — трохи наблизило б результати до людських.

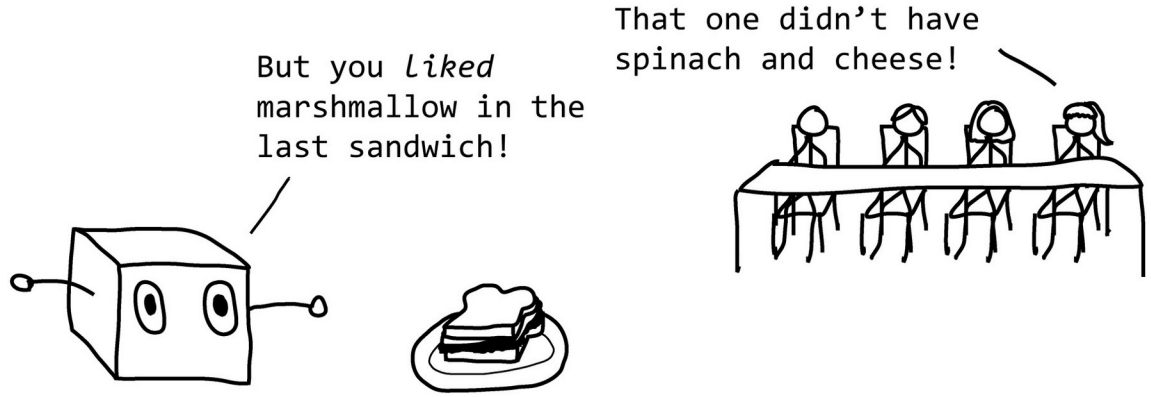




Тисячі ітерацій поступово виправляють мережу

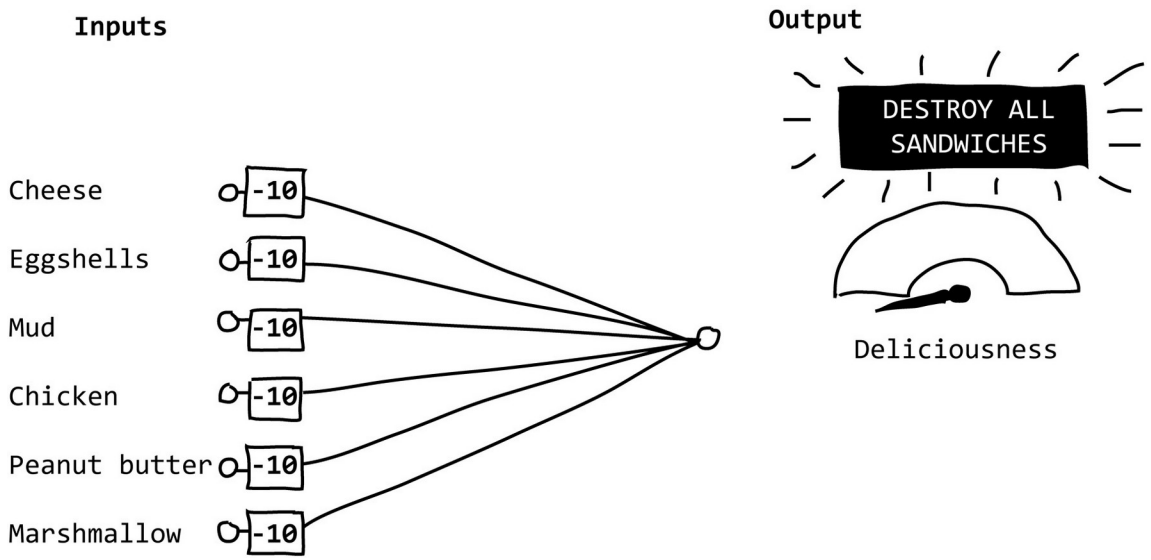
Зі щойно скоригованими вагами настав час нової ітерації. Нейромережа оцінює ще одну партію сендвічів, порівнює свої бали з людськими й знову змінює ваги. Після тисяч ітерацій та десятків тисяч сендвічів суддів уже нудить від усього цього, зате нейромережа працює значно краще.

Утім, на шляху прогресу чимало пасток. Як я вже казала, ця проста нейронна мережа знає лише, чи окремі інгредієнти загалом добрі або погані, й не здатна виробити тонке уявлення про вдалі поєднання. Для цього потрібна складніша структура з прихованими шарами клітин. Їй доведеться сформувати каральники та клітини делікатесних сендвічів.



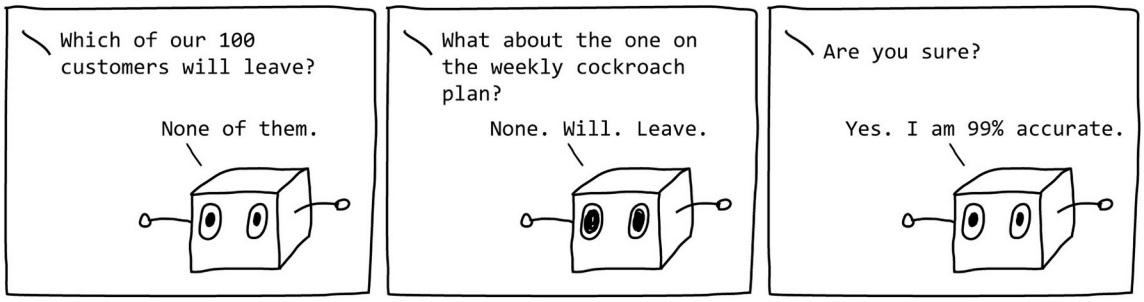
Дисбаланс класів навчає ШІ відкидати взагалі все

Ще одна пастка — **дисбаланс класів**. Пам'ятаєте, серед кожної тисячі сендвічів із діри смачними є лише кілька. Замість клопоту з вагами кожного інгредієнта та їхніми поєднаннями мережа може збагнути, що досягне точності 99,9 відсотка, якщо називатиме жахливим кожен сендвіч без винятку.



Щоб подолати дисбаланс класів, треба попередньо відфільтрувати навчальні сендвічі так, щоб смачних і жахливих було приблизно порівну. Навіть тоді нейромережа може не дізнатися про інгредієнти, яких зазвичай варто уникати, але які смакують у дуже специфічних умовах. Приклад — маршмелоу: жахливий із більшістю звичних складників, але чудовий у флафернатері (а можливо, ще з шоколадом і бананами). Якщо мережа не побачить флафернатерів під час навчання або бачитиме їх дуже рідко, то може вирішити, що досить точно працюватиме, відкидаючи все з маршмелоу.

> Проблеми дисбалансу класів постійно виникають у практичних застосуваннях — зазвичай коли ми просимо ШІ виявити рідкісну подію. Намагаючись передбачити, коли клієнти підуть із компанії, люди мають набагато більше прикладів тих, хто залишився, ніж тих, хто пішов. Тому ШІ ризикує скоротити шлях і вирішити, що всі клієнти залишаться назавжди. Та сама проблема є у виявленні шахрайських входів і хакерських атак, адже справжні атаки рідкісні. Про дисбаланс класів повідомляють і в медичній візуалізації, де серед сотень клітин шукають лише одну аномальну: ШІ спокушається отримати високу точність найкоротшим шляхом, оголосивши всі клітини здоровими. Астрономи теж стикаються з цим, бо чимало цікавих небесних подій рідкісні. Одна програма для виявлення сонячних спалахів з’ясувала, що майже стовідсоткової точності можна досягти, завжди прогнозуючи нуль спалахів, адже в навчальних даних вони траплялися дуже рідко.[2]



КОЛИ КЛІТИНИ ПРАЦЮЮТЬ РАЗОМ

Більшість нейронів виконують незрозумілу частину спільної роботи

На прикладі сортування сендвічів ми побачили, як шар клітин збільшує складність завдань, доступних нейронній мережі. Ми побудували клітину делікатесного сендвіча, що реагувала на поєднання м’яса й сиру, і клакерфлафер, який карав будь-який сендвіч за спробу поєднати курятину з маршмелоу. Але в нейромережі, що навчає себе сама, методом спроб і помилок налаштовуючи зв’язки між клітинами, визначити роботу кожної клітини зазвичай набагато важче. Завдання розподіляються між кількома клітинами, а для деяких узагалі складно чи неможливо зрозуміти, що саме вони роблять.

Щоб дослідити це явище, розгляньмо клітини повністю навченої нейромережі. Ця конкретна мережа, яку створили й навчили дослідники OpenAI,[3] прочитала літера за літерою понад вісімдесят два мільйони відгуків на товари Amazon і намагалася передбачити наступну літеру. Це ще одна рекурентна нейронна мережа, загалом така сама, як та, що створила жарти «тук-тук», смаки морозива й рецепти з розділів 1 і 2. Ця більша — у ній приблизно стільки нейронів, як у медузи. Ось кілька створених нею відгуків:

> Це чудова книжка, яку я рекомендував би всім, хто любить чудову історію персонажів і серію книжок.
>
> Я люблю цю пісню. Слухаю її знову й знову й ніколи не втомлююся. Вона так затягує. Люблю її!!
>
> Це найкращий засіб, яким я будь-коли чистив душову кабіну. Він не жирний, не здирає з води воду й не плямує білий килим. Я користуюся ним кілька років, і мені він добре підходить.
>
> Ці DVD з тренуваннями дуже корисні. Ними можна накрити всю свою дупу.
>
> Я купив це, думаючи, що воно добре підійде для гаража. У кого є багато озерної води? Я цілковито помилявся. Усе було просто й швидко. Нічний гризлі не зашкодив йому, а в нас це вже понад три місяці. Гості натхнені й справді ним тішаться. Мій тато його обожнює!

Ця нейромережа має окремий вхід для кожної літери чи розділового знака, які їй можуть трапитися (подібно до сортувальника сендвічів з окремим входом для кожного інгредієнта), і може озиратися на кілька попередніх літер та знаків. (Ніби оцінка сендвіча трохи залежала від останніх побачених сендвічів: можливо, система відстежує, чи не набридли нам сирні, й відповідно змінює оцінку наступного.) І замість єдиного виходу, як у сортувальника, нейромережа-відгукач має багато виходів — по одному на кожну літеру чи знак, який вона може вибрати як найімовірніший наступний. Якщо вона бачить послідовність «I own twenty eggbeaters and this is my very favorit», то найімовірнішим продовженням буде літера *e*.

За вихідними сигналами ми можемо подивитися на кожну клітину й побачити, коли вона «активна», а тоді обґрунтовано здогадатися про її функцію. У прикладі із сортувальником клітина делікатесного сендвіча була б активною, побачивши багато м’яса й сиру, та неактивною за наявності шкарпеток, кульок, арахісової пасти. Але більшість нейронів мережі відгуків Amazon будуть далеко не такими зрозумілими, як клітини делікатесних сендвічів і каральники. Більшість правил, винайдених мережею, будуть для нас незбагненними. Іноді ми здатні вгадати функцію клітини, але значно частіше гадки не маємо, що вона робить.

Ось активність однієї клітини алгоритму відгуків (2 387-ї), коли він створює текст (білий = активна, темний = неактивна):

> For me, this is one of the few albums of theirs I own that actually made me an instant classic pop fan. I also had a major problem with the audio with 10 new songs; the execution of the vocals and editing was awful. The next day, I was in a

> recording studio and I can’t tell you how many times I had to hit the play button to see where the song was going.

Ця клітина допомагає нейромережі передбачати наступні літери, але її функція загадкова. Вона реагує на певні літери чи їх поєднання, однак не так, щоб це мало для нас сенс. Чому її так захопили літери *um* у слові *album*, але не *al*? Що вона насправді робить? Це лише крихітний шматочок головоломки, який працює разом із багатьма іншими клітинами. Майже всі клітини нейромережі настільки ж загадкові. Сам уривок — відгук про музичний альбом: спершу він хвалить музику, а потім скаржиться на жахливі вокал і монтаж.

Нейрон настрою відрізняє похвалу від нищівної критики

Утім, час від часу трапляється клітина з упізнаваною роботою: наприклад, та, що активується між парою дужок, або дедалі сильніше активується зі збільшенням речення.[4] Дослідники, які навчали мережу відгуків, помітили одну зрозумілу клітину: вона реагувала на те, позитивний відгук чи негативний. Схоже, виконуючи завдання передбачити наступну літеру, мережа вирішила, що важливо з’ясувати, хвалити їй товар чи розносити його. Ось активація «нейрона настрою» в тому самому відгуку. Світлий колір означає високу активацію, тобто клітина вважає відгук позитивним:

> For me, this is one of the few albums of theirs I own that actually made me an instant classic pop fan. I also had a major problem with the audio with 10 new songs; the execution of the vocals and editing was awful. The next day, I was in a recording studio and I can’t tell you how many times I had to hit the “play” button to see where the song was going.

Відгук починається дуже позитивно, тож нейрон настрою сильно активований. Але посередині тон змінюється, й рівень активації клітини різко падає.

Ось іще один приклад його роботи. Активність низька, коли відгук нейтральний чи критичний, але швидко зростає, щойно клітина виявляє зміну настрою:

> The Harry Potter File, from which the previous one was based (which means it has a standard size liner) weighs a ton and this one is huge! I will definitely put it on every toaster I have in the kitchen since, it is that good. This is one of the best comedy movies ever made. It is definitely my favorite movie of all time. I would recommend this to ANYONE!

На Едгарі По нейрон настрою помиляється

Проте в інших видах тексту він гірше розпізнає тональність. Більшість людей не назвали б позитивним цей уривок із «Падіння дому Ашерів» Едгара Аллана По, а от конкретна нейромережа вважає його переважно позитивним:

> Overpowered by an intense sentiment of horror, unaccountable yet unendurable, I threw on my clothes with haste (for I felt that I should sleep no more during the night,) and endeavoured to arouse myself from the pitiable condition into which I had fallen, by pacing rapidly to and fro through the apartment.

Гадаю, фільм може сповнити вас сильним почуттям жаху й усе одно бути добрим — якщо саме це він і мав зробити.

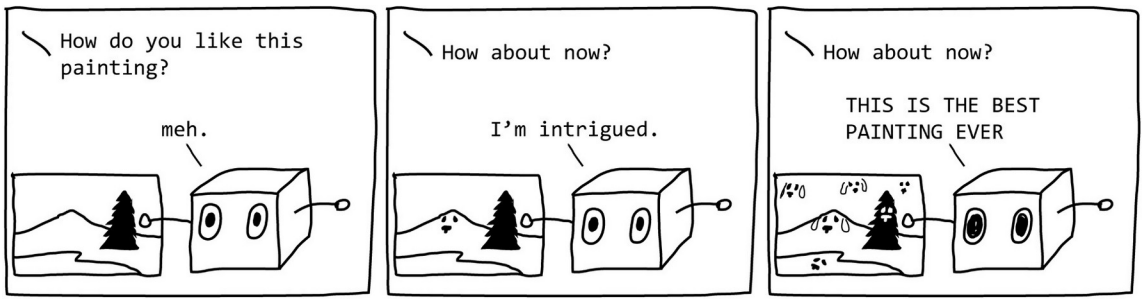
Знов-таки, для алгоритму створення чи аналізу тексту незвично мати клітину, поведінка якої настільки прозора, як у нейрона настрою. Те саме стосується й інших типів нейромереж — і це прикро, бо нам дуже хотілося б бачити, коли вони припускаються невдалих помилок, і вчитися на їхніх стратегіях.

Утім, в алгоритмах розпізнавання зображень клітини з упізнаваними функціями знайти трохи легше. Там входами є окремі пікселі зображення, а виходами — різні можливі класифікації (собака, кіт, жирафа, тарган тощо). Більшість таких алгоритмів мають між ними дуже багато шарів клітин — прихованих шарів. І в більшості є клітини чи їх групи, функції яких можна визначити, якщо правильно проаналізувати мережу. Можна подивитися на групи, що активуються від певних речей, або трохи змінювати вхідне зображення й перевіряти, які зміни найсильніше їх збуджують.

ГЛИБОКІ СНИ

DeepDream змінює зображення, доки нейрони не побачать усюди собак

> Зміна зображення заради сильнішого збудження нейронів — це техніка, якою створювали знамениті зображення Google DeepDream: нейромережа-розпізнавач перетворювала звичайні картини на пейзажі, повні психоделічних собачих морд і фантастичних нагромаджень арок та вікон.
>
> Щоб створити зображення DeepDream, беруть нейромережу, навчену щось розпізнавати, — наприклад собак. Потім обирають одну її клітину й поступово змінюють зображення так, щоб вона збуджувалася дедалі більше. Якщо клітину навчено розпізнавати собачі морди, вона сильніше активуватиметься, що більше ділянок зображення нагадуватимуть їх. Коли картинку доведуть до смаку клітини, та буде сильно спотворена й уся в собаках.



Найменші групи клітин, схоже, шукають краї, кольори й дуже прості текстури. Вони можуть повідомляти про вертикальні лінії, криві або зелені трав'янисті текстури. У наступних шарах більші групи клітин шукають набори країв, кольорів і текстур або прості деталі. Наприклад, дослідники Google проаналізували алгоритм розпізнавання GoogLeNET і з'ясували, що кілька груп клітин окремо шукали в тварин висячі та гострі вуха, допомагаючи відрізняти собак від котів.[5] Інші клітини збуджувалися від хутра або очних яблук.

«Операція на мозку» генератора прибирає штори, але не столи

Нейромережі, що створюють зображення, теж мають клітини з упізнаваною роботою. Ми можемо провести їм «операцію на мозку»: вимикати окремі клітини й дивитися, як змінюється створене зображення.[6] Група з MIT виявила, що деактивація клітин дає змогу прибирати елементи з картинок. Цікаво, що елементи, які нейромережа вважала «необхідними», вилучити було важче: наприклад, зі зображення конференц-зали легше забрати штори, ніж столи й стільці.

А тепер розгляньмо інший тип алгоритму — з ним ви, ймовірно, взаємодіяли безпосередньо, якщо користувалися функцією передбачення тексту на смартфоні.

ЛАНЦЮГИ МАРКОВА

Швидке передбачення наступного слова з дуже короткою пам'яттю

Ланцюг Маркова — це алгоритм, здатний розв'язувати багато тих самих задач, що й рекурентна нейронна мережа (РНМ), яка створювала рецепти, смаки морозива, відгуки Amazon і назви метал-гуртів у цій книжці. Як і РНМ, він дивиться на минуле (наприклад, уже вжиті в реченні слова чи погоду минулого тижня) й передбачає, що найімовірніше станеться далі.

Ланцюги Маркова легші за більшість нейромереж і швидше навчаються. Саме тому функція передбачення тексту у смартфонах зазвичай використовує ланцюг Маркова, а не РНМ.

Однак зі збільшенням пам'яті ланцюг Маркова експоненційно стає дедалі громіздкішим. Приміром, пам'ять більшості систем передбачення тексту охоплює лише три-п'ять слів. Натомість РНМ можуть пам'ятати сотні слів, а завдяки **LSTM** (довгій короткочасній пам'яті) та згортковим прийомам — навіть більше. У розділі 2 ми побачили, як коротка пам'ять змушує РНМ втрачати важливі відомості. З ланцюгами Маркова так само.

Три слова пам’яті замикають пісні Disney в нескінченній петлі

Я навчила ланцюг Маркова на наборі пісень Disney, скориставшись навчуваною клавіатурою з передбаченням тексту.[7] Навчання тривало лише кілька секунд, а не кілька хвилин, як у РНМ. Але цей ланцюг має пам’ять завдовжки три слова. Тобто він пропонує слова, які вважає найімовірнішими на основі трьох попередніх слів пісні. Коли я змусила його створити пісню, щоразу обираючи лише найімовірнішу пропозицію, він видав ось це:

> Море)
> під водою)
> під водою)
> під водою)
> під водою)
> під водою)
> під водою)

Він не знає, скільки разів співати «під водою», бо не пам’ятає, скільки разів уже це заспівав.

Якщо почати з перших слів пісні з «Красуні і Чудовиська» («Історія стара, як час»), він швидко знову застрягає.

> Історія стара, як час
> пісня стара, як час
> пісня стара, як час
> пісня стара, як час

У кількох куплетах «Красуні і Чудовиська» за словами «історія стара, як час» ідуть слова «пісня стара, як рима». Але коли цей ланцюг бачить фразу «стара, як», він не знає, який саме з двох куплетів зараз пише.

Я можу витягти його з пастки, щоразу вибираючи *друге* за ймовірністю слово. Тоді він пише таке:

> Цілий світ
> яскравий юний господар
> ти з усім
> гадаєш ти дивом
> силами
> і я занепав духом
> самотнє пекельне полум’я темний бік

А якщо щоразу вибирати третє за ймовірністю слово:

> Ти думаєш я можу відкритися
> де ми побачимо що ти відчуваєш

> усе це мої мрії будуть моїми
> щось там було раніше
> їй буде кращий час

Так значно цікавіше, але сенсу майже немає. А пісні — й поезія — досить поблажливі до граматики, структури та зв’язності. Якщо дати ланцюгу Маркова інший набір даних, його вади стануть іще очевиднішими.

Ось ланцюг, навчений на списку першоквітневих розіграшів: на кожному кроці він обирає найімовірніше наступне слово. (Розділових знаків він не пропонував, тому поділ на рядки додала я.)

> Зняти дверну ручку з дверей і поставити її назад задом наперед тихо
> Нічого не робити весь день із ко чийогось оголошення в газеті про гаражний розпродаж у когось про неминучий розіграш
> Тоді нічого не робити весь день із ко чийогось оголошення в газеті про гаражний розпродаж у…

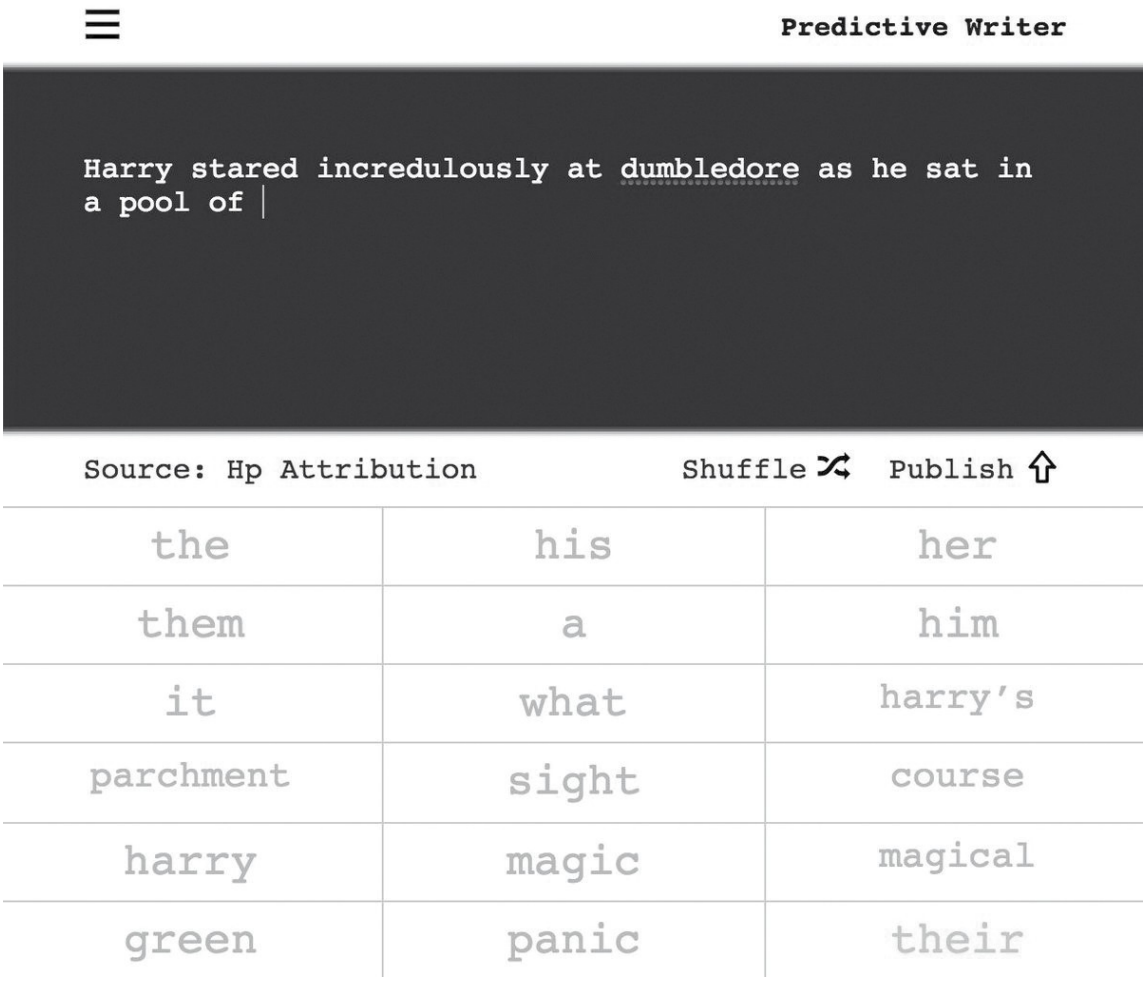
Ланцюг Маркова для передбачення тексту навряд чи зможе поговорити з клієнтом або написати історію для нового квесту у відеогрі (люди сподіваються, що колись РНМ робитимуть і те, й інше). Зате він уміє пропонувати ймовірні наступні слова для конкретного навчального набору.

Botnik лишає вибір дивних слів людині

Наприклад, команда Botnik навчає ланцюги Маркова на різних наборах (книжках про Гаррі Поттера, епізодах *«Зоряного шляху»*, відгуках Yelp тощо) й пропонує слова людям-письменникам. Несподівані варіанти алгоритму часто допомагають спрямувати тексти в дивно сюрреалістичні боки.

Замість дозволити ланцюгу з короткою пам’яттю самостійно вибрати наступне слово, я можу попросити його вигадати кілька варіантів і показати їх мені — так само, як працює передбачення тексту під час набору повідомлення.

Ось як виглядає взаємодія з одним із навчених ланцюгів Botnik — цю модель навчали на книжках про Гаррі Поттера:



А ось кілька нових першоквітневих розіграшів, які я написала за допомогою передбачення тексту навченого ланцюга Маркова:

> Покладіть гранули пластикової плівки собі на губи.
> Складіть кухонну раковину у формі курячої голови.
> Візьміть світну паличку й удайте, ніби чхаєте на даху.
> Перетворіть сидіння унітаза на штани, а тоді попросіть машину попісяти.

Для порівняння я також використала складнішу й вимогливішу до даних РНМ для створення першоквітневих розіграшів. Тут РНМ генерувала весь розіграш разом із пунктуацією. Але частка людської творчості все одно залишилася: мені довелося перебрати всі її пропозиції та знайти найсмішніші.

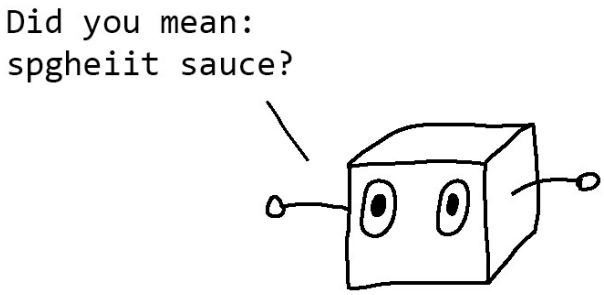
> Зробіть їжу в офісному комп'ютері когось.
> Сховайте весь вхід до вашої офісної будівлі, якщо в неї лише один вхід.
> Наліпіть рухливі оченята на чиюсь комп'ютерну мишу, щоб вона не працювала.
> Поставте миску із сумішшю M&M's, Skittles і Reese's Pieces.
> Покладіть штани й черевики у свій диспенсер для льоду.

Власні друкарські помилки можуть переслідувати вас місяцями

Подібні експерименти можна провести з передбаченням тексту в більшості застосунків для повідомлень. Почніть із «Я народився...» або «Жили собі...» й далі натискайте запропоновані телефоном слова — отримаєте дивний текст просто з нутрощів алгоритму машинного навчання. Оскільки новий ланцюг Маркова навчається порівняно швидко й легко, текст буде особисто вашим. Системи передбачення та автовиправлення в телефоні оновлюються в міру того, як ви друкуєте, навчаючись на вашому письмі. Саме тому одна помилка може переслідувати вас іще дуже довго.

Інтернет навчив Google Docs виправляти правильні слова на неправильні

Google Docs, можливо, став жертвою подібного ефекту: користувачі повідомляли, що автовиправлення змінювало правильне англійське «a lot» на неправильне «alot» і пропонувало розмовне «gonna» замість «going». Google використовував контекстне автовиправлення, яке сканувало інтернет, вирішуючи, що запропонувати.[8] Перевага контекстного виправлення в тому, що воно бачить помилки, які утворюють справжні слова (скажімо, англійське «gong» замість «going»), і додає нові слова, щойно ті стають поширеними. Але кожен користувач інтернету знає: часте вживання рідко збігається з формально «правильним», якого ми очікуємо від автовиправлення в текстовому редакторі. Google не коментував саме ці помилки, але після скарг користувачів вони зазвичай зникали.



ВИПАДКОВІ ЛІСИ

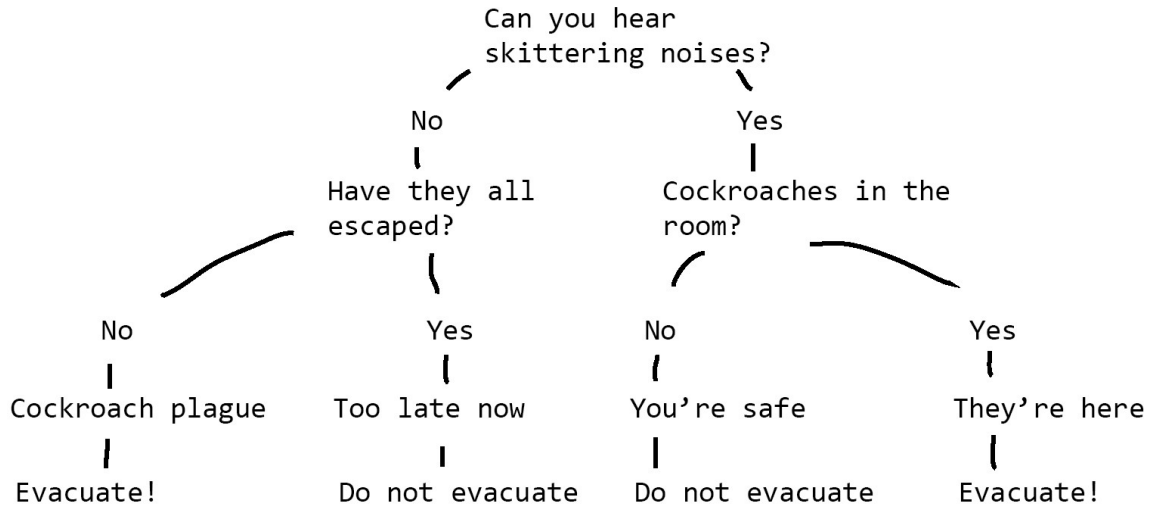
Алгоритм випадкового лісу — це різновид машинного навчання, який часто застосовують для прогнозування та класифікації: наприклад, щоб передбачати поведінку покупців, рекомендувати книжки або оцінювати якість вина на основі багатьох вхідних даних.

Дерево рішень вирішує, чи евакуювати тарганячу ферму

Щоб зрозуміти ліс, почнімо з дерев. Випадковий ліс складається з окремих одиниць — дерев рішень. **Дерево рішень** — це, по суті, блок-схема, яка на основі наявної інформації приводить до певного результату. І, що приємно, дерева рішень справді трохи схожі на перевернуті дерева.

На наступній сторінці — приклад гіпотетичного дерева рішень про те, чи слід евакуювати гігантську тарганячу ферму.

Дерево відстежує, як ми використовуємо відомості (зловісні звуки, наявність тарганів), щоб вирішити, як діяти. Так само як рішення щодо сендвічів стають витонченішими зі збільшенням кількості клітин у нейромережі, більший розмір дерева дає змогу враховувати ситуацію з тарганами тонше.



Якщо на фермі підозріло тихо, але таргани не втекли, можуть існувати інші (можливо, ще тривожніші) пояснення, крім «усі померли». У більшому дереві можна було б запитати, чи лежать десь мертві таргани, наскільки розумними вони вважаються й чи не були таємничим чином саботовані машини для їх розчавлювання.

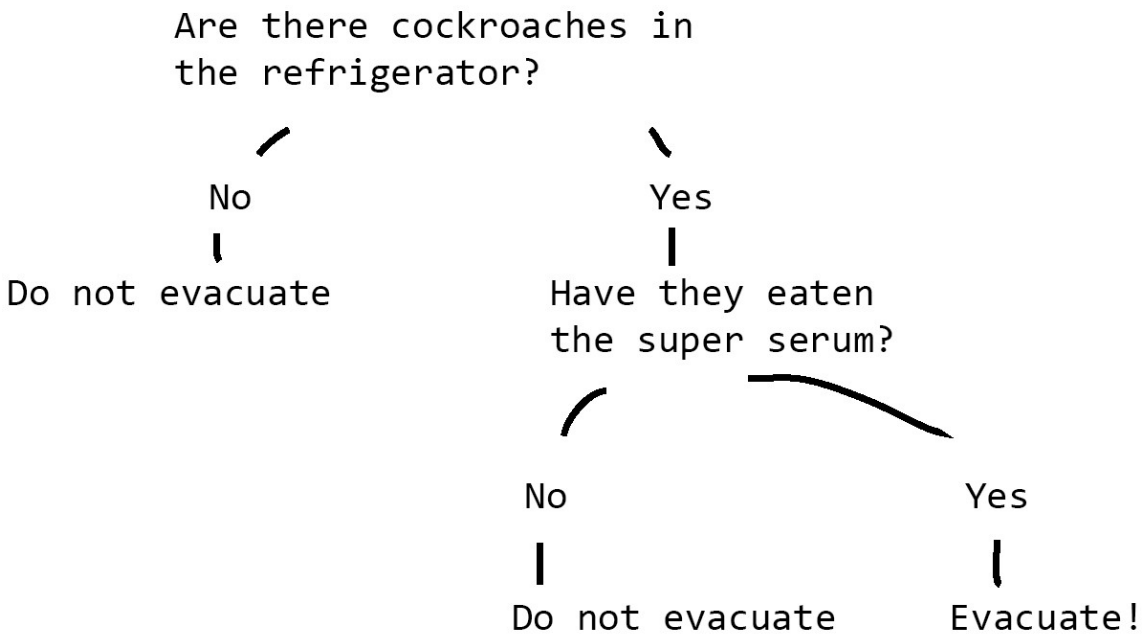
З великою кількістю входів і варіантів дерево може стати неймовірно складним (або, мовою програмування глибокого навчання, дуже глибоким). Воно здатне розростися так, що охопить кожен можливий вхід, рішення й результат із навчального набору, але тоді працюватиме лише в конкретних навчальних ситуаціях. Інакше кажучи, перенавчиться на навчальних даних. Людина-експерт могла б хитро побудувати величезне дерево, яке уникає перенавчання й обробляє більшість рішень, не чіпляючись за конкретні, ймовірно несуттєві дані. Наприклад, якщо минулого разу під час втечі тарганів

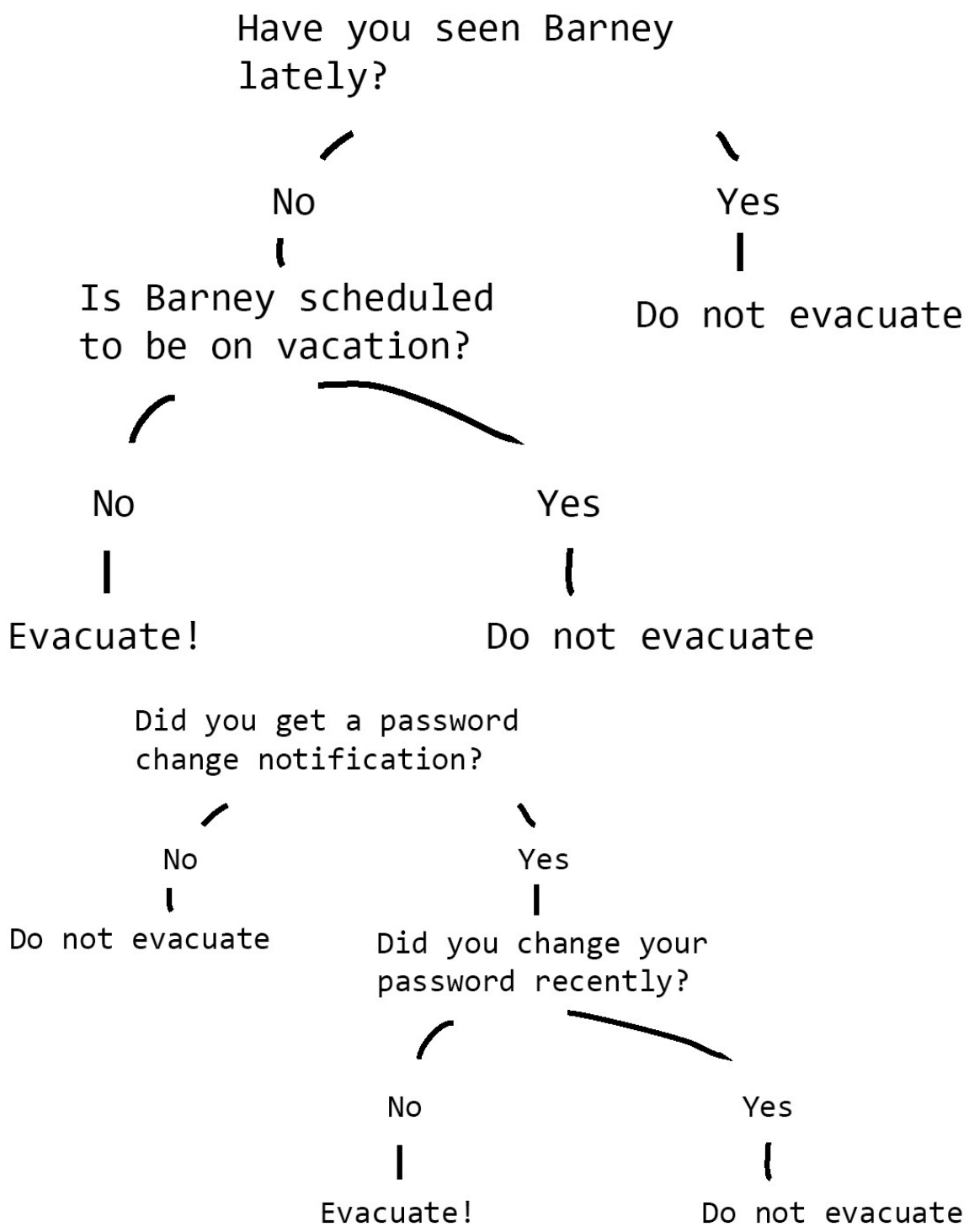
було хмарно й прохолодно, людина досить розумна, щоб не вважати, ніби така сама погода обов’язково пов’язана з наступною втечею.

Безліч слабких дерев голосує точніше за одне велике

Альтернатива ретельному людському конструюванню величезного дерева — метод випадкового лісу. Подібно до того як нейромережа спробами й помилками налаштовує зв’язки між клітинами, алгоритм випадкового лісу налаштовує сам себе. Ліс складається з безлічі крихітних (тобто неглибоких) дерев, кожне з яких розглядає малу частину інформації та ухвалює кілька простих рішень. Під час навчання кожне дерево з’ясовує, на які дані звертати увагу й до якого результату доходити. Окреме рішення, ймовірно, буде слабким, бо спирається на дуже обмежені відомості. Але якщо всі дерева об’єднають висновки й проголосують за остаточний результат, то разом будуть набагато точнішими. (Те саме відбувається з людьми: коли вони вгадують кількість кульок у банці, окремі відповіді можуть дуже помилятися, а їх середнє значення, ймовірно, буде близьким до істини.) Дерева випадкового лісу можуть об’єднувати рішення з найрізноманітніших питань і створювати точну картину приголомшливо складних ситуацій. Одне з недавніх застосувань, приміром, перебирало сотні тисяч геномних закономірностей, щоб визначити, який вид худоби спричинив небезпечний спалах кишкової палички.[9]

Якби ми доручили випадковому лісу ситуацію з тарганами, кілька його дерев могли б мати такий вигляд:





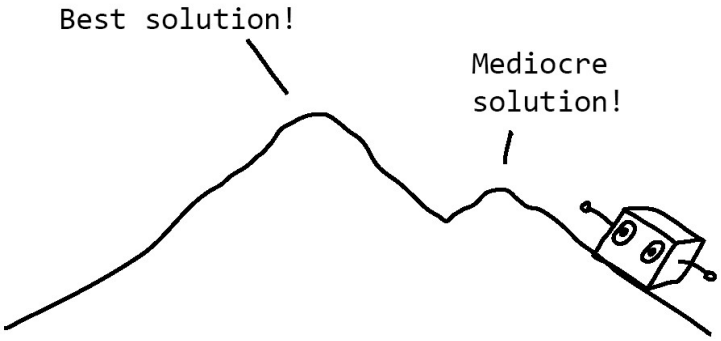
Кожне окреме дерево бачить лише крихітний шматок ситуації. Відсутність Барні може мати цілком розумне пояснення — можливо, він просто взяв лікарняний. А якщо таргани не з’їли суперсироватку, це ще не означає, що ми в безпеці. Можливо, вони взяли її зразки й саме зараз варять величезну партію — достатню для 1,7 мільярда тарганів у комплексі.

Та дерева поєднують свої здогади, і коли Барні загадково зник, сироватка пропала, а ваш пароль незбагненно змінився, евакуація може бути розсудливим рішенням.

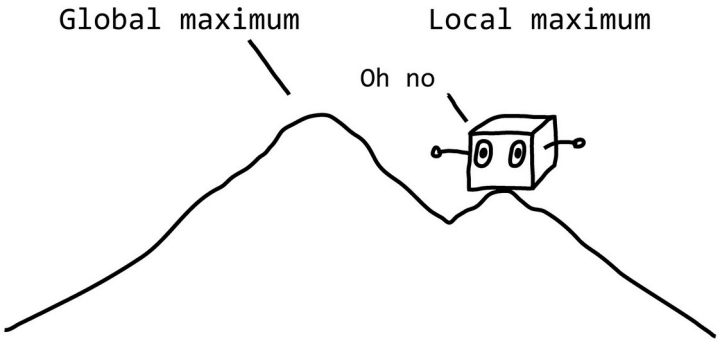
ЕВОЛЮЦІЙНІ АЛГОРИТМИ

Підйом на пагорб може привести не на найвищу вершину

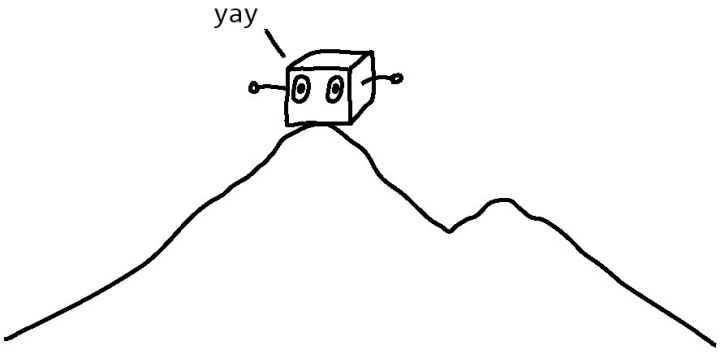
ШІ уточнює своє розуміння, вгадуючи добре рішення, а потім перевіряючи його. Усі три розглянуті вище алгоритми методом спроб і помилок вдосконалюють власну структуру, створюючи таку конфігурацію нейронів, ланцюгів і дерев, яка найкраще розв’язує задачу. Найпростіші методи завжди рухаються в напрямку поліпшення. Їх часто називають **сходженням на пагорб**, якщо треба максимізувати число (скажімо, очки в Super Mario Bros.), або **градієнтним спуском**, якщо треба його мінімізувати (наприклад, кількість тарганів-утікачів). Та просте наближення до мети не завжди дає найкращий результат. Щоб побачити пастки сходження, уявіть, що стоїте десь на горі в густому тумані й шукаєте її найвищу точку.



Простий алгоритм сходження завжди поведе вас угору. Але залежно від стартової точки ви можете зупинитися на найнижчій вершині — **локальному максимумі**, а не на найвищій, тобто **глобальному максимумі**.

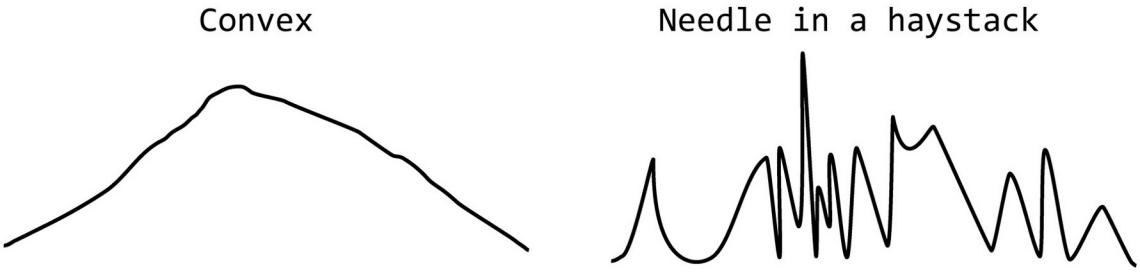


Тому існують складніші методи спроб і помилок, які змушують досліджувати більше ділянок гори: наприклад, здійснити кілька пробних походів у різних напрямках, перш ніж визначати найперспективніші місця. Так можна дослідити гору ефективніше.



Мовою машинного навчання гора називається **простором пошуку**: десь у цьому просторі лежить мета (на горі є вершина), і ви намагаєтеся її знайти. Деякі простори **опуклі** — базовий алгоритм сходження щоразу приведе на вершину. Інші набагато прикріші. Найгірші — так звані **задачі голки в копиці сіна**, де ви майже не маєте уявлення, наскільки близькі до найкращого рішення, аж доки випадково не натрапите на нього. Пошук простих чисел — приклад такої задачі.

Простором пошуку алгоритму може бути будь-що. Наприклад, форми деталей, із яких складається ходячий робот. Або всі можливі ваги нейронної мережі, де «вершиною» є ваги, що допомагають розпізнавати відбитки пальців чи обличчя. Або можливі конфігурації випадкового лісу, серед яких треба знайти ту, що добре передбачає улюблені книжки покупця — чи потребу евакуювати тарганячу фабрику.



Як ми вже з’ясували, базовий пошук на кшталт сходження на пагорб або градієнтного спуску мало допоможе, якщо простір можливих конфігурацій нейромережі не надто опуклий. Тому дослідники іноді звертаються до інших, складніших методів спроб і помилок.

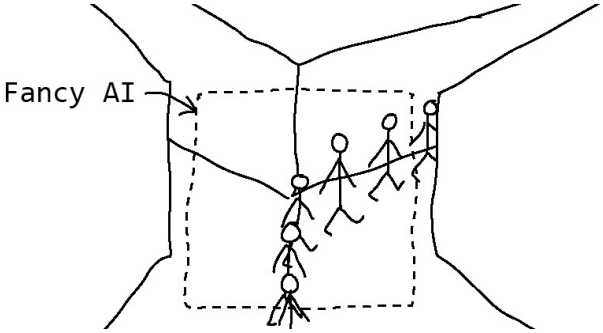
Еволюція перетворює «вгадай і перевір» на покоління рішень

Одна зі стратегій надихається еволюцією. Наслідувати її цілком логічно: зрештою, що таке еволюція, як не процес «вгадай і перевір» через покоління? Якщо істота чимось відрізняється від сусідів, і ця відмінність підвищує її шанси вижити й розмножитися, вона передасть корисні ознаки наступному поколінню. Риба, яка плаває трішки швидше за інших представників виду, частіше тікатиме від хижаків, а через кілька поколінь її швидкі нащадки можуть стати поширенішими за нащадків повільніших риб. Еволюція — надзвичайно могутній процес. Вона розв’язала незліченні задачі руху та обробки інформації, навчилася добувати їжу із сонячного світла й гідротермальних джерел, а також світитися, літати й ховатися від хижаків, удаючи пташиний послід.

В **еволюційних алгоритмах** кожне потенційне рішення подібне до організму. У кожному поколінні найуспішніші рішення виживають і розмножуються, мутуючи або схрещуючись з іншими, щоб породити відмінних — і, сподіваємося, кращих — нащадків.

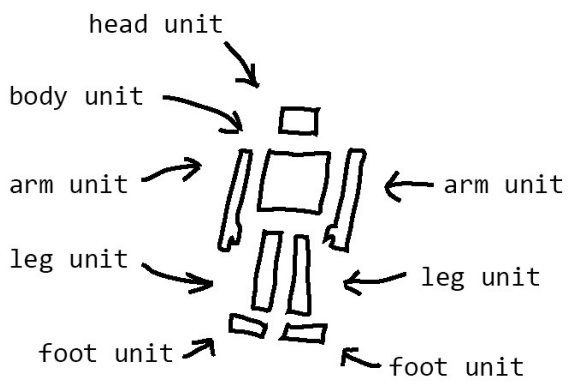
Геном робота визначає форму тіла й рухи

Якщо ви колись билися над складною задачею, думка про кожне потенційне рішення як живу істоту — яка їсть, парується й робить усяке інше — може приголомшити. Розгляньмо конкретний приклад. Скажімо, ми розв’язуємо задачу керування натовпом: маємо коридор, що розгалужується, і хочемо створити робота, який спрямовуватиме людей в один прохід або інший.



Спершу визначаємо складники, які еволюційний алгоритм може змінювати: що в роботі має бути сталим, а з чим програма вільна гратися. Варіативність можна дуже обмежити — зафіксувати будову тіла й дозволити змінювати лише рухи. Або дозволити алгоритму збудувати тіло повністю з нуля, починаючи з випадкових грудок. Припустімо, власники будівлі заради науково-фантастичної естетики наполягають на людиноподібному дизайні. Жодної безладної купи повзучих блоків (а саме так зазвичай виглядають створіння

еволюційного алгоритму, якщо дати йому повну свободу). Навіть у базовій людиноподібній формі можна змінювати багато чого, але спростімо: нехай алгоритм варіює розмір і форму кількох основних частин тіла, кожна з яких має простий діапазон руху. Еволюційною мовою це **геном** робота.




```
Robot Genome

Body part dimensions:
   Head unit: length, width, height
   Body unit: length, width, height
   ...
Behaviors:
   Default behavior
   When human present
   When human moves left
   When human moves right
   ...
```

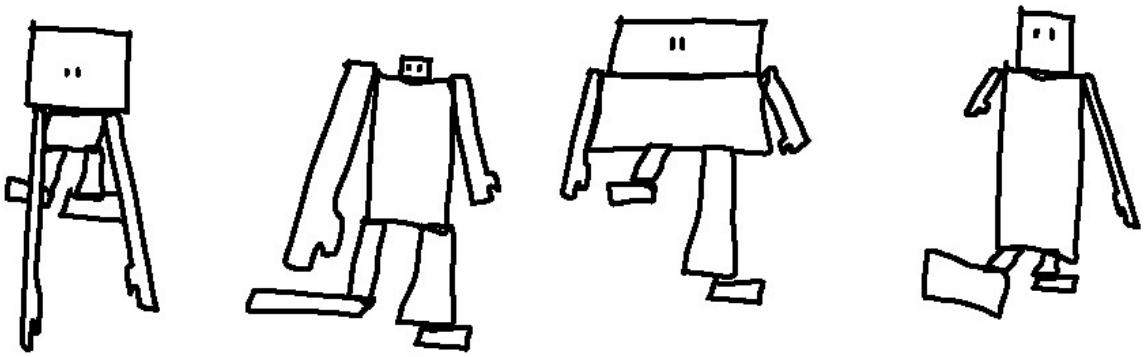
Функція пристосованості й симуляція задають мету еволюції

Далі треба визначити задачу так, щоб отримати єдине число для оптимізації. Еволюційною мовою це **функція пристосованості**: одне число, яке показує, наскільки окремий робот придатний для нашого завдання. Ми будуємо робота, що спрямовує людей одним із двох коридорів, тож спробуймо мінімізувати кількість тих, хто піде ліворуч. Що ближче це число до нуля, то вища пристосованість.

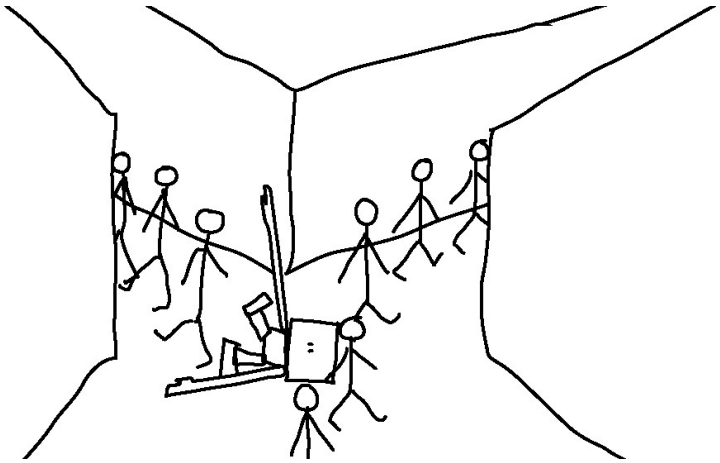
Також потрібна симуляція, бо ми не збираємося замовляти тисячі роботів або наймати людей, щоб вони тисячі разів ходили коридором. (Відмова від справжніх людей — ще й питання безпеки; причини незабаром стануть очевидні.) Нехай це буде змодельований коридор у світі зі змодельованими гравітацією, тертям та іншою фізикою. І, звісно, потрібні змодельовані люди зі змодельованою поведінкою: ходьбою, полем зору, скупченнями, різними фобіями, мотиваціями та рівнями поступливості. Сама симуляція — дуже складна задача, тому просто скажімо, що ми вже її розв’язали. (Примітка: у справжньому машинному навчанні так легко не буває.)

> Зручний спосіб отримати готову симуляцію для навчання ШІ — скористатися відеогрою. Частково тому так багато дослідників навчають ШІ грати в Super Mario Bros. або старі ігри Atari: це маленькі й швидкі програми, здатні перевіряти різні навички розв’язання задач. Але, як і люди-гравці, ШІ схильний знаходити й використовувати помилки ігор. Докладніше — у розділі 5.

Дозволяємо алгоритму випадково створити перше покоління роботів. Вони… дуже випадкові. Типове покоління містить сотні роботів, кожен з іншим тілом.

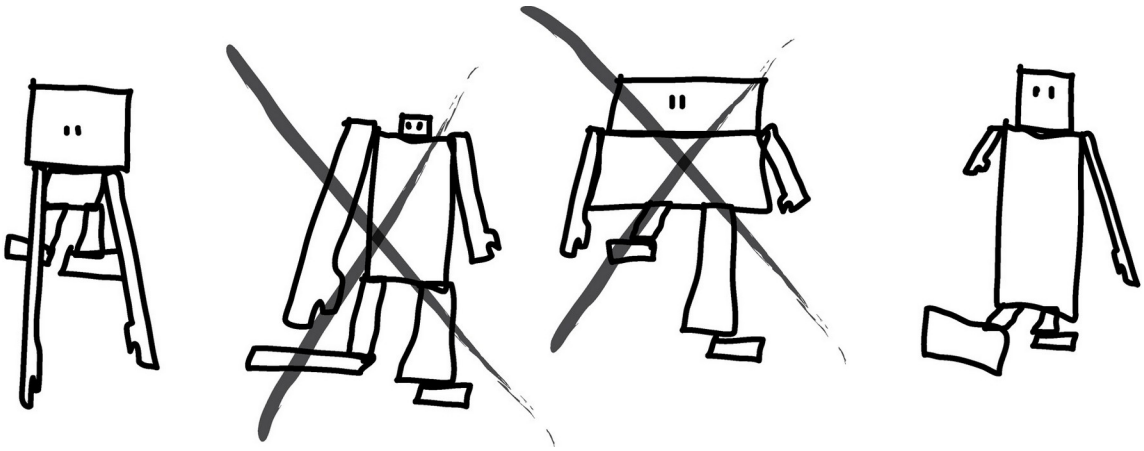


Тоді випробовуємо кожного окремо в змодельованому коридорі. Виходить погано. Люди проходять повз, поки роботи плюхаються на підлогу й безпорадно вимахують кінцівками. Може, один падає трохи лівіше за інших і частково перекриває коридор, а кілька боязкіших людей через це обирають правий. Цей робот отримує трішки кращий бал.

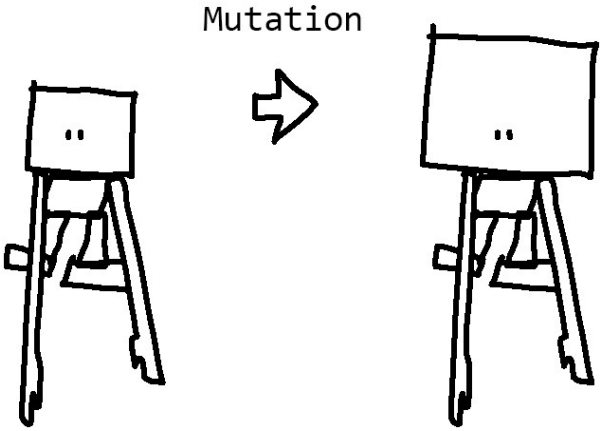


Добір, мутація і схрещування створюють наступне покоління

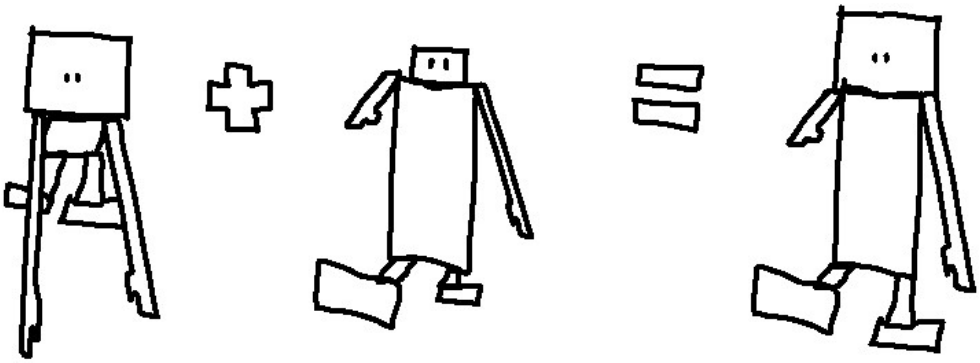
Настав час будувати наступне покоління. Спершу виберемо роботів, які виживуть і розмножаться. Можна зберегти лише найкращого, але тоді популяція стане надто одноманітною й ми не перевіримо інші конструкції, які могли б виявитися кращими, якби еволюція дістала нагоду їх підправити. Тож збережемо кількох найкращих, а решту відкинемо.



Далі маємо багато варіантів того, як уцілілі роботи розмножуватимуться. Вони не можуть просто копіювати себе, бо ми хочемо еволюції до чогось кращого. Один варіант — **мутація**: узяти випадкового робота й випадково щось у ньому змінити.



Інший варіант — **схрещування**: два роботи створюють нащадка, що є випадковим поєднанням обох батьків.



Також треба вирішити, скільки нащадків може мати кожен робот (чи найуспішніші повинні мати їх найбільше?), хто з ким може схрещуватися (і чи

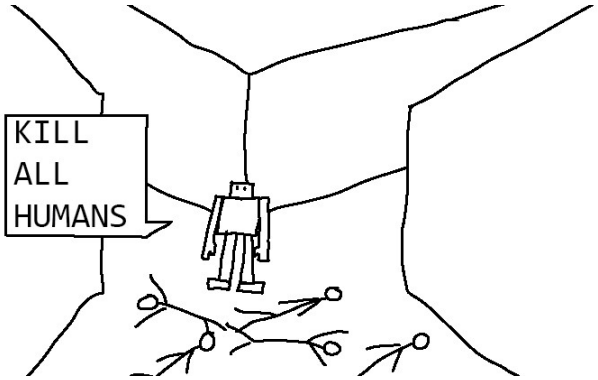
використовуємо ми схрещування взагалі), а також чи замінимо всіх загиблих нащадками або кількома заново створеними випадковими роботами. Налаштування цих параметрів — важлива частина побудови еволюційного алгоритму, й іноді складно вгадати, які саме **гіперпараметри** спрацюють найкраще.

Створивши нове покоління, ми знову запускаємо цикл і випробовуємо його в симуляції. Тепер більше роботів падає ліворуч, бо вони походять від першого, ледь успішного робота.

Погана ціль еволюціонує робота, який убиває всіх

Через багато поколінь виникають виразні стратегії керування натовпом. Коли роботи вчаться стояти, первісне «впасти ліворуч і трохи заважати» еволюціонує в «стояти в лівому коридорі й заважати ще більше». З'являється й інша стратегія — «енергійно показувати праворуч». Але жодна ще не розв'язує задачу ідеально: кожен робот усе одно пропускає чимало людей ліворуч.

Ще через багато поколінь з'являється робот, який надзвичайно добре не пускає людей ліворуч. На жаль, через прикрий збіг обставин знайдене ним рішення — «убити всіх». Технічно воно працює, бо ми наказали лише мінімізувати кількість людей, які входять у лівий коридор.



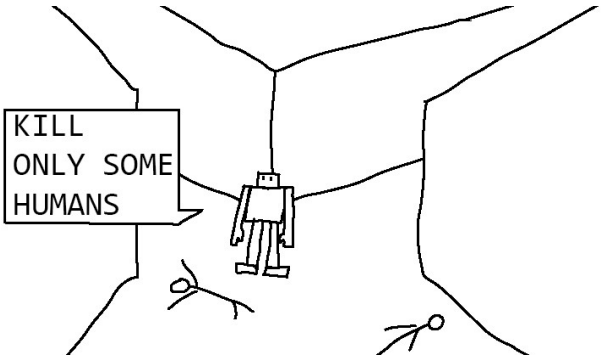
Через проблему з функцією пристосованості еволюція спрямувала алгоритм до непередбаченого рішення. Невдалі скорочення трапляються в машинному навчанні постійно, хоча зазвичай не настільки драматичні. (На щастя, у реальності «вбити всіх людей» переважно дуже непрактично. Висновок: не давайте автономним алгоритмам смертельної зброї.) Саме *тому* в уявному експерименті ми використали змодельованих людей, а не справжніх.

Доведеться почати знову — цього разу з функцією, яка не мінімізує кількість людей у лівому проході, а максимізує кількість тих, хто обирає правий.

Зміна функції пристосованості перевиховує роботів-убивць

Насправді можна трохи криваво скоротити шлях: не починати все заново, а лише змінити функцію пристосованості. Адже крім убивства людей роботи опанували багато корисного: стояти, помічати людей і моторошно рухати руками. Коли нова функція почне максимізувати кількість уцілілих, які входять у правий коридор, роботи *мають* швидко відмовитися від убивчих звичок. (Нагадаю, повторне використання рішення з іншої, але пов’язаної задачі називається перенесенням навчання.)

Отже, починаємо з групи роботів-убивць і нишком підміняємо їм функцію пристосованості. Раптом убивства перестають працювати, а роботи не знають чому. Ба більше, той, хто вбивав найгірше, тепер опиняється на вершині, бо кільком його жертвам, що кричали, вдалося втекти правим коридором. Упродовж наступних поколінь роботи стрімко стають дедалі гіршими в убивстві.

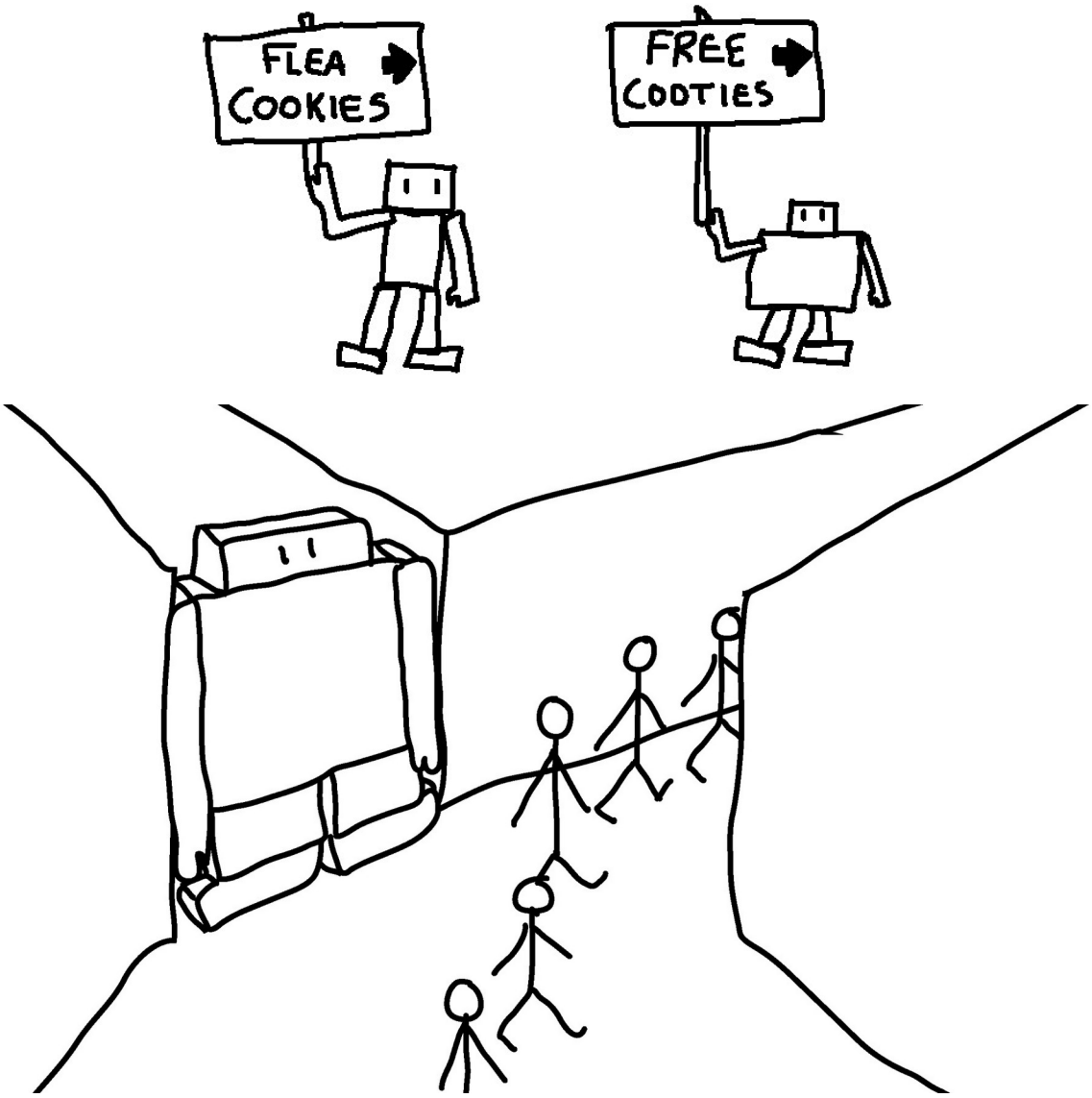


Зрештою, можливо, вони лише виглядатимуть так, ніби *хочуть* вас убити, — і цього вистачить, щоб більшість людей злякалася й пішла праворуч. Починаючи з популяції роботів-убивць, ми обмежуємо шлях, яким, найімовірніше, піде еволюція. Якби почали заново, могли б отримати роботів, які стоять у кінці правого коридору й підкликають людей, або навіть роботів, чиї руки перетворилися на таблички БЕЗПЛАТНЕ ПЕЧИВО. (Утім, робота з печивом еволюціонувати важко, бо частково правильний напис не спрацює, а винагородити рішення, яке лише наближається до мети, буде складно. Інакше кажучи, це рішення-голка в копиці сіна.)

Найімовірніший ідеальний робот виявляється звичайними дверима

Якщо відкласти роботів-убивць, найімовірніше еволюціонував би той, хто «падає й заважає», стаючи дедалі дошкульнішою перепоною. (Падати дуже легко, тому якщо еволюційний робот може розв’язати задачу падінням, то

зазвичай саме так і зробить.) Цим шляхом ми можемо дійти до робота, який ідеально спрямовує сто відсотків людей праворуч, не вбивши жодної. Виглядає він так:



Так, ми еволюціонували… двері.

Це ще одна риса ШІ: іноді він може бути безглуздо складною заміною здорового розуміння задачі.

Еволюційні алгоритми проєктують бампери, білки й маховики

Еволюційні алгоритми використовують для розвитку всіляких конструкцій, не лише роботів. Автомобільні бампери, що поглинають силу під час зминання; білки, які зв’язуються з іншими медично корисними білками; маховики, що обертаються саме так, як треба, — усі ці задачі люди розв’язували

еволюційними алгоритмами. Геном не обов’язково має описувати фізичний об’єкт. Можна взяти автомобіль або велосипед із незмінною конструкцією й еволюціонувати програму керування. Як я вже згадувала, геномом можуть бути навіть ваги нейронної мережі або будова дерева рішень. Різні види машинного навчання часто поєднують саме так, використовуючи сильні сторони кожного.

Дивлячись на величезне розмаїття життя, що виникло на Землі завдяки еволюції, ми отримуємо уявлення про масштаб можливостей віртуальної еволюції, прискореної в багато разів. Як справжня еволюція створила дивовижно складних істот і дала їм змогу користуватися найдивнішими й найвужчими джерелами їжі, так еволюційні алгоритми продовжують дивувати й тішити нас винахідливістю. Звісно, іноді вони можуть виявитися *надто* творчими — це ми побачимо в розділі 5.

ГЕНЕРАТИВНО-ЗМАГАЛЬНІ МЕРЕЖІ (GAN)

ШІ здатні творити із зображеннями неймовірне: перетворювати літній пейзаж на зимовий, створювати обличчя вигаданих людей або робити з фотографії чийогось кота кубістичну картину. Такі ефектні інструменти генерації, змішування й фільтрування зображень зазвичай є роботою **GAN (генеративно-змагальних мереж)**. Це підвид нейромереж, але він заслуговує на окрему згадку. На відміну від інших типів машинного навчання в цьому розділі, GAN з’явилися нещодавно: їх представили Ян Ґудфеллоу та інші дослідники Монреальського університету лише у 2014 році.[10]

Генератор і дискримінатор навчаються, обманюючи одне одного

Ключова риса GAN — це насправді два алгоритми в одному, два суперники, які навчаються, випробовуючи одне одного. Перший, **генератор**, намагається імітувати вхідний набір даних. Другий, **дискримінатор**, намагається відрізнити імітацію генератора від справжніх даних.

Щоб зрозуміти користь цього способу навчання генератора зображень, розгляньмо гіпотетичний приклад. Припустімо, ми хочемо навчити GAN створювати зображення коней.

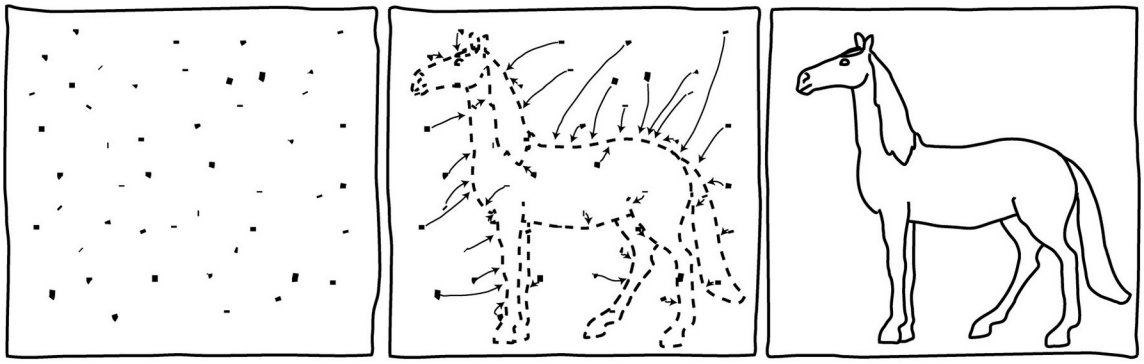
GAN малює переконливих коней лише у дуже простому світі

Насамперед знадобиться багато прикладів фотографій коней. Якщо на всіх буде той самий кінь у тій самій позі (може, ми ним одержимі), GAN навчатиметься швидше, ніж на величезній різноманітності мастей, ракурсів та освітлення. Завдання можна спростити й однаковим однотонним тлом. Інакше

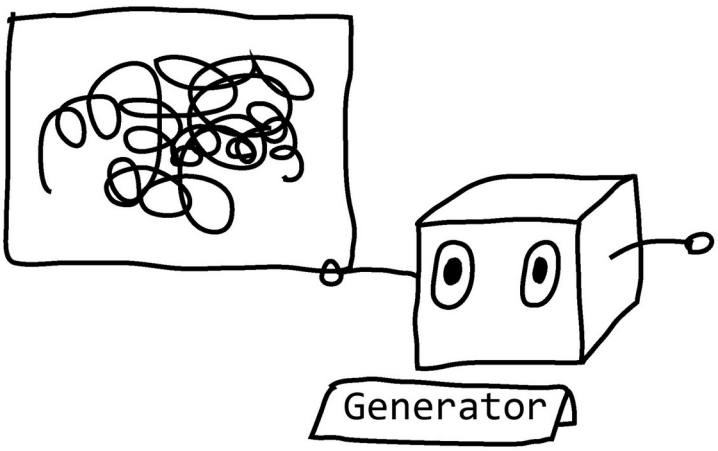
мережа довго вчитиметься, коли і як малювати паркани, траву та паради. Більшість GAN, здатних створювати фотореалістичні обличчя, квіти й страви, отримували дуже обмежені й однорідні набори: приміром, лише котячі морди або миски рамену, сфотографовані тільки згори. GAN, навчена лише на голівках тюльпанів, може переконливо малювати тюльпани, але нічого не знатиме про інші квіти чи навіть про те, що тюльпани мають листя й цибулини. GAN, яка створює фотореалістичні людські портрети, не знатиме, що нижче шиї або на потилиці, а навіть що людські очі можуть заплющуватися. Отже, наша мережа коней матиме більше шансів, якщо її світ буде дуже простим і вона бачитиме лише коней у профіль на білому тлі. (Зручно, що приблизно цим і обмежуються мої здібності до малювання.)

Генератор перетворює випадковий шум на різних коней

Тепер, коли набір даних готовий (або ми принаймні його уявили), можна навчати обидві частини GAN. Ми хочемо, щоб генератор розглянув набір кінських зображень і вивів правила, за якими створюватиме подібні. Технічно ми просимо його перетворювати випадковий шум на зображення коней. Так він створюватиме не одну картинку, а нового коня для кожного випадкового шумового візерунка.



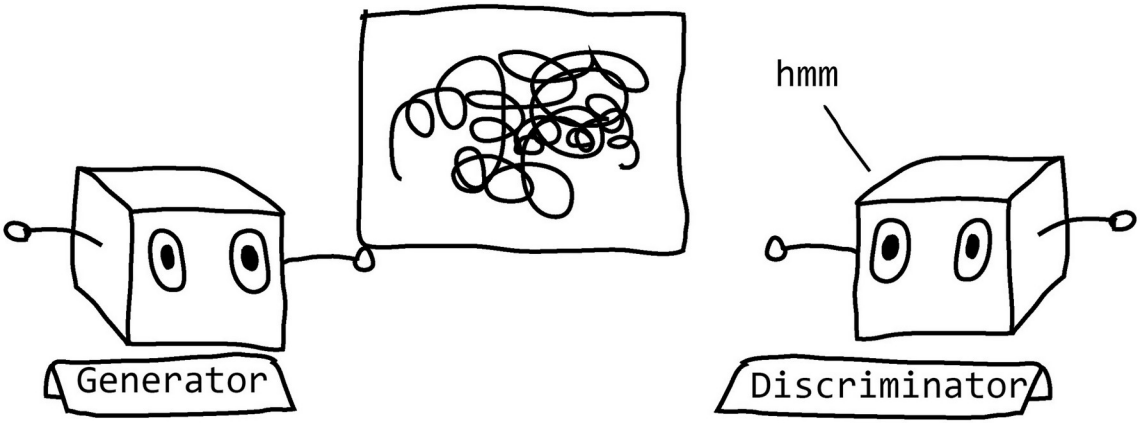
На початку навчання генератор іще не знає жодних правил малювання коней. Він бере випадковий шум і робить із ним щось випадкове. Наскільки йому відомо, саме так і малюють коня.



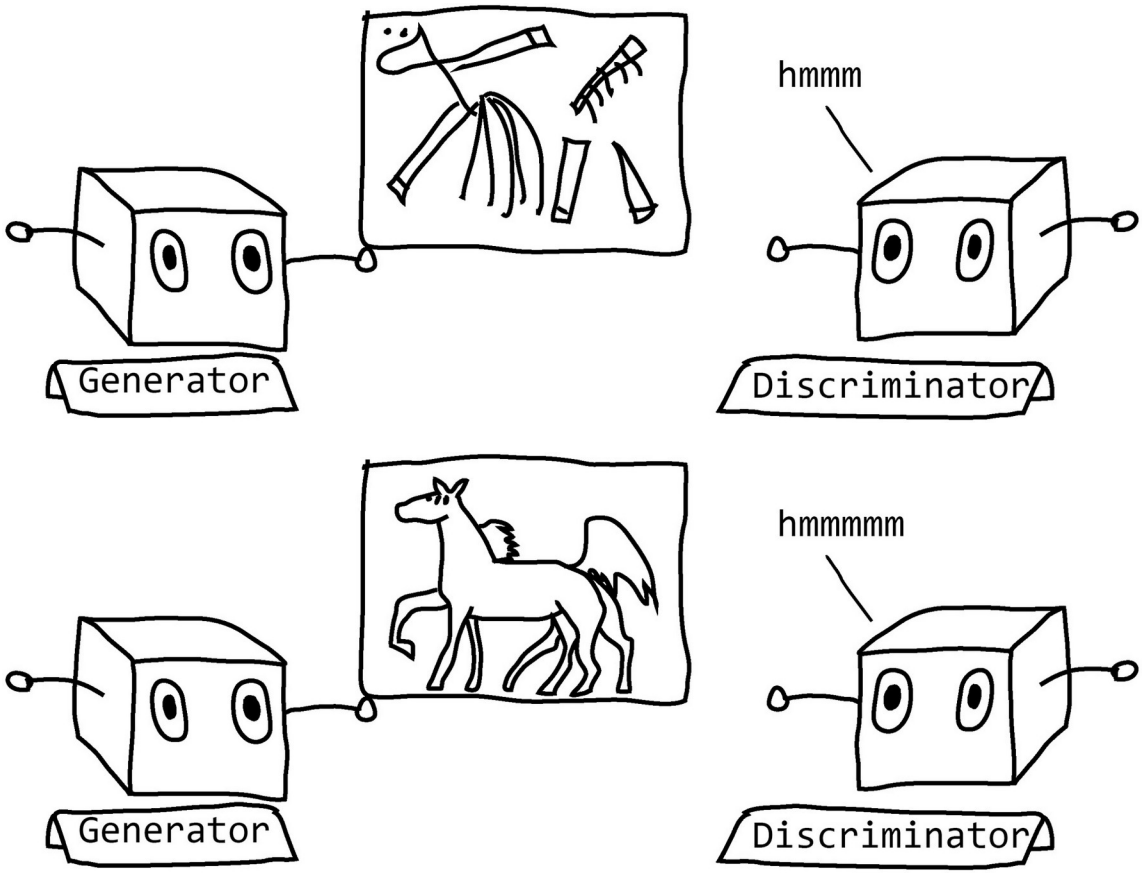
Недосвідченого дискримінатора здатна обдурити ледь конеподібна пляма

Як дати генератору корисний відгук про його жахливі малюнки? Алгоритмові потрібен відгук у вигляді числа — кількісна оцінка, яку можна поліпшувати. Корисною метрикою був би відсоток випадків, коли малюнок настільки добрий, що схожий на справжнього коня. Людина легко це оцінить: ми добре відрізняємо розмазане хутро від коня. Але навчання потребує багатьох тисяч малюнків, тому людина не може перевірити всі. Та й на цьому етапі вона була б надто суворою: подивилася б на дві карлючки генератора й обидві оцінила як «не кінь», навіть якщо одна ледь помітно конеподібніша. Якщо повідомляти генератору лише частоту, з якою він обдурив людину, той ніколи не знатиме про поступ, бо не обдурить її жодного разу.

Тут і з’являється дискримінатор. Його робота — розглядати малюнки й вирішувати, чи це справжні коні з навчального набору. На початку він майже такий самий нікчемний, як генератор: ледь відрізняє карлючки від справжнього. Тому ледь помітно конеподібній плямі може таки вдатися його обдурити.

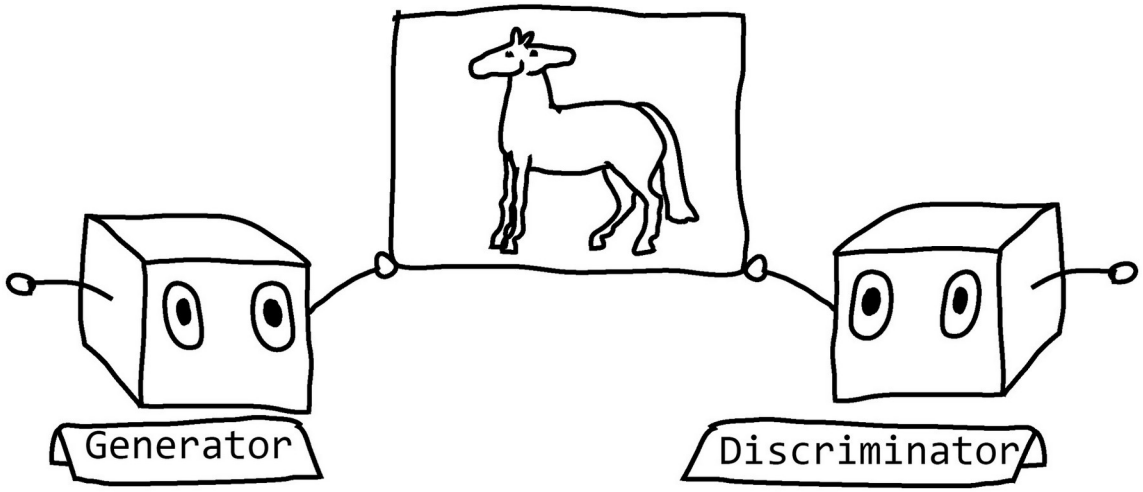


Методом спроб і помилок і генератор, і дискримінатор стають кращими.



GAN проводить тест Тюрінга, де сама є суддею й учасницею

У певному сенсі GAN використовує генератор і дискримінатор для тесту Тюрінга, у якому вона водночас і суддя, і учасниця. Сподіваємося, що наприкінці навчання вона створюватиме коней, які обдурили б і людину.



Суперечливі цілі змушують GAN балансувати між наслідуванням та новизною

Іноді люди проєктують GAN, які не намагаються точно відтворити навчальний набір, а прагнуть створити щось «подібне, але інше». Наприклад, дослідники зробили GAN для абстрактного мистецтва, але не хотіли нудних копій навчальних робіт. Вони налаштували дискримінатор так, щоб мистецтво було схоже на навчальні дані, але його не можна було віднести до конкретної категорії. Завдяки двом дещо суперечливим цілям GAN втрималася на межі між відповідністю та новаторством.[11] Її зображення стали популярними: люди-судді навіть оцінили їх вище, ніж картини людей.

ЗМІШУВАННЯ, ПОЄДНАННЯ І СПІЛЬНА РОБОТА

Ми дізналися, що GAN досягає мети, поєднуючи два алгоритми: один створює зображення, другий їх класифікує.

Насправді багато систем ШІ складаються з комбінацій інших, більш спеціалізованих алгоритмів машинного навчання.

Seeing AI об’єднує окремі моделі для опису світу вголос

Наприклад, застосунок Microsoft Seeing AI розроблено для людей із порушеннями зору. Залежно від вибраного «каналу» він уміє:

- розпізнавати вміст сцени й описувати її вголос,
- читати текст, показаний камері смартфона,
- розпізнавати номінали грошей,
- визначати людей та їхні емоції,
- знаходити й сканувати штрихкоди.

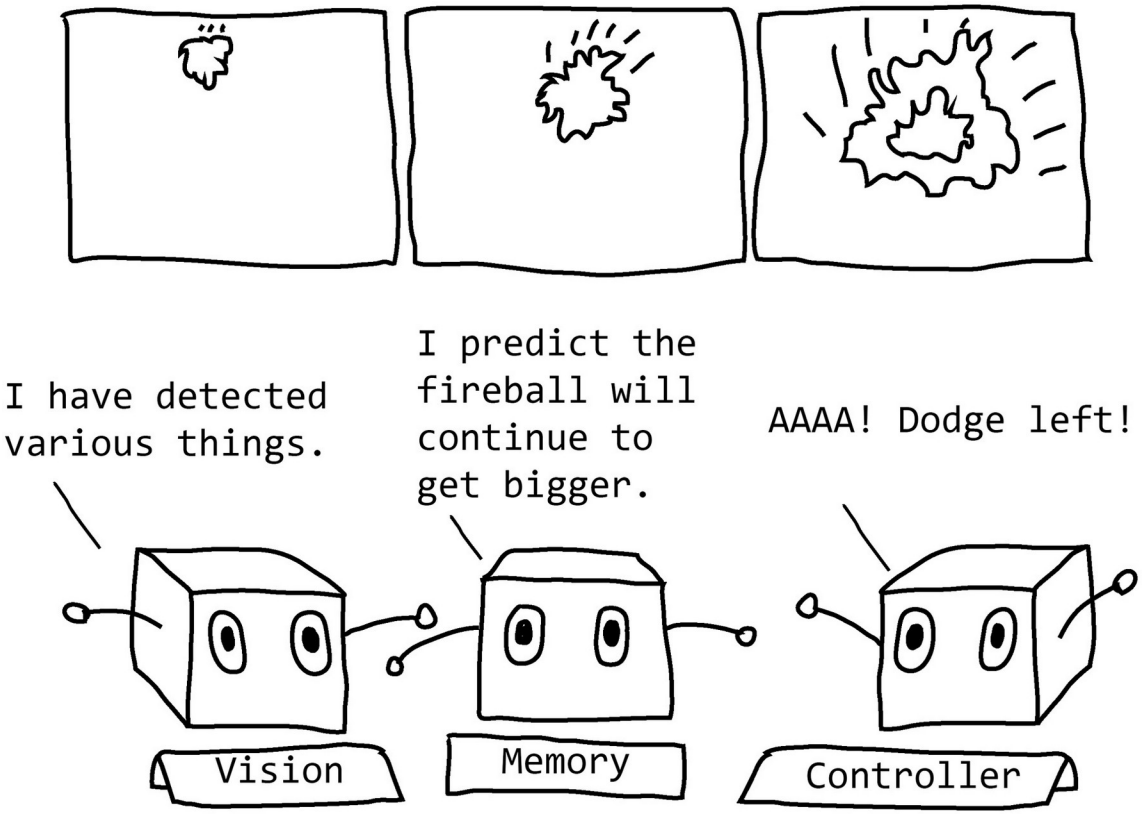
Кожну з цих функцій — разом із критично важливим синтезом мовлення — найімовірніше, забезпечує окремо навчений ШІ.

Три алгоритми створюють, стилізують і називають картини

Митець Ґреґорі Шатонскі використав три алгоритми машинного навчання для створення картин у проєкті *It’s Not Really You* («Насправді це не ти»).[12] Один алгоритм навчався створювати абстрактне мистецтво, другий перетворював його роботи на різні мальовничі стилі. Нарешті, митець використав розпізнавач зображень, щоб дати картинам назви на кшталт *«Барвистий салат», «Торт-поїзд»* і *«Піца, що сидить на камені»*. Остаточна робота стала співпрацею багатьох алгоритмів, спланованою й організованою митцем.

Зір, пам’ять і контролер разом ухиляються від вогняних куль у Doom

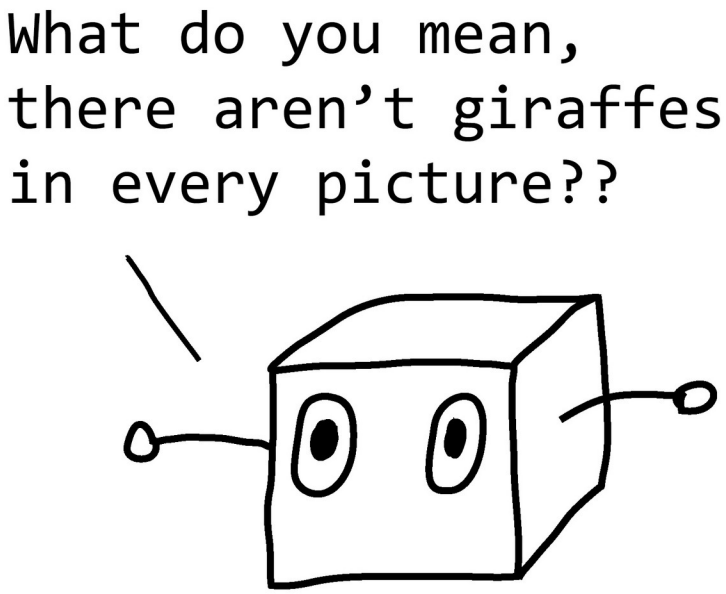
Іноді алгоритми інтегровані ще тісніше й одночасно виконують кілька функцій без втручання людини. Наприклад, дослідники Девід Га та Юрґен Шмідгубер за допомогою еволюції навчили натхнений людським мозком алгоритм проходити один рівень комп’ютерної гри Doom.[13] Система складалася з трьох алгоритмів, які працювали разом. Модель зору відповідала за сприйняття подій у грі: чи видно вогняні кулі, чи близько стіни? Вона перетворювала двовимірне зображення з пікселів на ознаки, які вирішила відстежувати. Друга, модель пам’яті, намагалася передбачати майбутнє. Так само як текстові РНМ у цій книжці дивляться на минуле й прогнозують наступну літеру чи слово, ця РНМ аналізувала попередні миті гри й передбачала наступну. Якщо кілька митей тому вогняна куля рухалася ліворуч, то на наступному зображенні вона, ймовірно, лишиться там, але трохи далі ліворуч. Якщо куля збільшувалася, то, напевно, збільшуватиметься й далі (або влучить у гравця й спричинить величезний вибух). Нарешті, третім алгоритмом був контролер, який вирішував, що робити. Чи варто ухилитися ліворуч від вогняної кулі? Можливо, це добра ідея.



Отже, три частини разом бачили вогняні кулі, розуміли, що ті наближаються, й ухилялися. Дослідники обрали форму кожного підалгоритму так, щоб вона найкраще відповідала конкретному завданню. Це логічно, бо з розділу 2 ми знаємо: алгоритми машинного навчання найкраще працюють із дуже вузькими задачами. Вибір правильної форми алгоритму або поділ проблеми на завдання для підалгоритмів — ключовий спосіб, яким програмісти можуть закласти успіх.

У наступному розділі ми розглянемо інші способи спроєктувати ШІ успішно — або не дуже.

РОЗДІЛ 4. Він старається!



П’ять способів дати ШІ завдання, з яким він не впорається

Досі ми говорили про те, як ШІ навчається розв’язувати задачі, які задачі йому вдаються та як він може привести нас до загибелі. Зосередьмося ще трохи на загибелі — випадках, коли рішення на основі ШІ є жахливим способом розв’язати реальну проблему. Наслідки бувають від легкого роздратування до дуже серйозних. У цьому розділі ми поговоримо, що стається, коли ШІ погано розв’язує задачу, і що з цим робити. Це може трапитися, коли ми:

- дали йому надто широку задачу,
- не надали достатньо даних, щоб він зрозумів, що відбувається,
- випадково дали дані, які його заплутали або змарнували його час,
- навчали його для завдання, набагато простішого за те, що трапилося в реальному світі, або
- навчали в умовах, які не відображали реального світу.

НАДТО ШИРОКА ЗАДАЧА

Це може бути знайоме з розділу 2, де ми розглядали задачі, придатні для ШІ. Як показав провал M, помічника Facebook на основі ШІ, якщо задача надто широка, алгоритмові буде важко давати корисні відповіді.

StyleGAN малює реалістичних людей, але котів із зайвими очима

У 2019 році дослідники Nvidia — компанії, що виробляє обчислювальні рушії, широко використовувані для ШІ, — навчили GAN (двокомпонентну змагальну нейромережу, про яку йшлося в розділі 3) під назвою StyleGAN створювати зображення людських облич.[1] StyleGAN упоралася вражаюче добре: її обличчя були фотореалістичними, якщо не зважати на дрібниці на кшталт різних сережок і трохи безглуздого тла. Та коли команда натомість навчила StyleGAN на фотографіях котів, мережа створила тварин із зайвими кінцівками, очима й дивно спотвореними мордами. Набір людських зображень складався з облич, сфотографованих спереду, а котячий містив котів із різних ракурсів: вони ходили, згорталися клубком або нявкали в камеру. StyleGAN довелося навчатися на великих планах, фотографіях кількох котів і навіть кадрах із людьми — це було надто багато для одного алгоритму. Важко повірити, що фотореалістичних людей і спотворених котів створив той самий базовий алгоритм. Але що вужче завдання, то розумнішим здається ШІ.

БІЛЬШЕ ДАНИХ, БУДЬ ЛАСКА

Згаданий алгоритм StyleGAN і більшість інших ШІ у цій книжці навчаються на прикладах. Маючи достатньо прикладів — котячих імен, малюнків коней, вдалих рішень за кермом чи фінансових прогнозів, — вони можуть виявити закономірності й наслідувати побачене. Але без достатньої кількості прикладів алгоритмові бракуватиме інформації, щоб зрозуміти, що відбувається.

Вісім смаків морозива не пояснюють, що таке морозиво

Доведімо це до крайності й подивімося, що буде, якщо навчати нейромережу вигадувати нові смаки морозива на неймовірно малій кількості прикладів. Дамо їй лише вісім:

> Шоколад
> Ваніль
> Фісташка
> Сліди лося
> Арахісова паста зі шматочками шоколаду
> М’ятне з шоколадними шматочками

> Блакитний місяць
> Шампанське, бурбон і ваніль із айвово-золотисто-малиновим завитком та зацукрованим імбиром

Це, безперечно, добрі класичні смаки. Людина, побачивши такий список, найімовірніше, зрозуміє, що це морозиво, і зможе додати ще кілька варіантів. Скаже, наприклад, «полуниця». Або «масляний пекан із чорничним завитком». Вона здатна на це, бо знає, що таке морозиво й які смаки в ньому зазвичай трапляються. Знає, як пишуться ці слова й навіть у якому порядку їх ставити (англійське *Mint Chocolate Chip*, приміром, ніколи не перетворюється на *Chip Chocolate Mint*). Вона знає, що полуниця існує, а ґланґбері — ні.

Для ненавченої мережі літери, пробіли й переноси — лише числа

Але ненавчена нейромережа, якій я даю той самий список, не має жодних таких знань. Вона не знає, що таке морозиво чи навіть англійська мова. Не розуміє, що голосні відрізняються від приголосних, а літери — від пробілів і переносів рядка. Корисно показати набір так, як його бачить нейромережа: кожна літера, пробіл і розділовий знак перетворені на одне число.

> 3;8;15;3;15;12;1;20;5;24;22;1;14;9;12;12;1;24;16;9;19;20;1;3;8;9;15;24;13;15;15
> ;19;5;0;20;18;1;3;11;19;24;16;5;1;14;21;20;0;2;21;20;20;5;18;0;3;8;9;16;24;13;9
> ;14;20;0;3;8;15;3;15;12;1;20;5;0;3;8;9;16;24;2;12;21;5;0;13;15;15;14;24;3;8;1;1
> 3;16;1;7;14;5;0;2;15;21;18;2;15;14;0;22;1;14;9;12;12;1;0;23;9;20;8;0;17;21;9;14
> ;3;5;26;7;15;12;4;5;14;0;18;1;19;16;2;5;18;18;25;0;19;23;9;18;12;0;1;14;4;0;3;1
> ;14;4;9;5;4;0;7;9;14;7;5;18;

Завдання нейромережі — з’ясувати, наприклад, коли може з’явитися символ 13 (літера *m*). Двічі він стоїть після символу 24 (переносу рядка), але одного разу — після символу 0 (пробілу). Чому? Ми, звісно, цього прямо не пояснювали. Погляньмо й на символ 15 (літеру *o*). Іноді він з’являється двічі поспіль (обидва рази після символу 13), а ще кілька разів — лише один. Знову ж таки, чому? Даних замало, щоб ШІ це зрозумів. А оскільки літера *f* узагалі не трапилася у вхідному наборі, нейромережа не призначила їй числа. Наскільки їй відомо, *f* не існує. Хоч би як старалася, вона нізащо не вигадає *toffee*, *coffee* чи *fudge*.

Утім, нейромережа *дуже* старається й дещо таки створює. Вона дізнається, що голосні й пробіли (символи 1, 5, 9, 15, 21 і 0) трапляються часто, тому на ранньому етапі навчання її результат має такий вигляд:

> aaaoo aaaaaaaaoalnat ia eain l e ee r r e r er n r en d edeedr ed
> d nrd d edi r rn n d e e eer d r e d d dd dr rr er r r n e ri d edAe
> eri diedd rd eder r edder dnrr dde er ne r dn nend n dn rnndr
> eddnr re rdre rdd e r e e dnrddrr rdd r

Через єдину дуже довгу назву смаку в навчальному наборі мережа не відразу розуміє, як часто використовувати символ 24 — перенос рядка. Але згодом її результат починає розділятися на окремі «смаки»:

> tahnlaa aa otCht aa itonaC hi aa gChoCe ddidddddRe dCAnddddriddrni dedweiliRee aataa naa ai
> tttCuat

Фаза моторошної корови: мукання, шипіння й перші впізнавані слова

Ще довше доводиться чекати на перші впізнавані слова з набору: алгоритм починає запам’ятовувати окремі комбінації літер і проходить якусь «фазу моторошної корови», коли мукає, шипить і вряди-годи пирхає:

> MoooootChopooopteeCpp
> BlpTrrrks
> Bll Monoooooooooooo
> Pnstchhhhhh
> MoooosTrrrkssssss
> PeniautBuut tChppphippphppihpppi
> Moonillaaaaal
> Pnnillaaa
> Buee Moooo

У міру навчання смаки стають упізнаванішими:

> Moint Chooolate Chip
> Peanut Butter Chip
> Pistchhio
> Bue Moo
> Moose Trrack
> Psenutcho
> Vanilla
> MintCcooolate Chhip
> Psstchhio
> Chaampgne Booouorr VanillaWith QciiG-Golddni aspberrrr ndirl AndCandiiddnngger

Ідеальне зазубрювання восьми прикладів не є творчим узагальненням

Мережа вже може дослівно копіювати деякі вхідні смаки, бо запам’ятовує дедалі довші вдалі послідовності символів. Якщо навчати її ще, вона зрештою ідеально відтворить усі вісім назв. Але ми прагнули не цього. Запам’ятати вхідні приклади — не те саме, що навчитися створювати нові смаки. Інакше кажучи, алгоритм не зміг узагальнити.

На 2011 прикладах мережа нарешті починає винаходити

З набором належного розміру нейромережа просувається значно краще. Коли я навчила її на 2 011 смаках (набір усе ще малий, але вже не сміховинно), ШІ нарешті став винахідливим. Він створив нові смаки з наведеного нижче списку, а також ті, що були в розділі 2; жодного з них не було у вихідному наборі.

> Копчене масло
> Бурбонова олія
> Печений буряк із пеканом
> Випасена олія
> Зелений чай із кокосом
> Шоколад з імбиром, лаймом та Oreo
> Морквяне пиво
> Червоний мед
> Лайм із кардамоном
> Шоколад, Oreo, олія + тофі
> Молочний імбирний шоколад із перцем-горошком

Великі компанії вже мають дані, а решті доводиться їх збирати

Отже, для навчання ШІ більше даних зазвичай означає кращий результат. Саме тому нейромережа відгуків Amazon із розділу 3 навчалася на приголомшливих вісімдесяти двох мільйонах відгуків. І тому, як ми дізналися з розділу 2, безпілотні автомобілі навчаються на мільйонах кілометрів реальних доріг і мільярдах кілометрів симуляцій, а стандартні набори для розпізнавання зображень на кшталт ImageNet містять багато мільйонів картинок.

Але де взяти стільки даних? Якщо ви Facebook або Google, ці величезні набори, можливо, вже у вас. Наприклад, Google накопичив стільки пошукових запитів, що зміг навчити алгоритм передбачати кінець речення, коли ви починаєте друкувати в пошуку. (Недолік навчання на даних справжніх користувачів: підказки можуть бути сексистськими та/або расистськими. А іноді просто дивними.) В епоху великих даних потенційні навчальні дані для ШІ — цінний актив.

Якщо готових даних немає, їх доведеться якось зібрати. Один дешевий варіант — краудсорсинг, якщо проєкт достатньо веселий чи корисний, щоб людям не набридло. Так збирали набори для розпізнавання тварин на фотопастках, китових голосів і навіть змін температури в дельті данської річки. Дослідники, які створюють інструмент для підрахунку зразків під

мікроскопом, можуть просити користувачів надсилати розмічені дані, щоб поліпшувати майбутні версії.

Один «помічник» завантажує до набору весь каталог костюмів

Та іноді краудсорсинг працює гірше, і я звинувачую в цьому людей. Наприклад, я збирала список костюмів на Гелловін і просила добровольців вписати в онлайн-форму все, що вони могли придумати. Алгоритм почав створювати такі костюми:

> Спортивний костюм
> Сексуальний костюм для переляку
> Загальна конструкція переляку

Проблема була в тому, що хтось, мабуть бажаючи допомогти, завантажив увесь асортимент магазину костюмів. («Ким ти маєш бути?» — «О, я Чоловічий костюм IT класу люкс — стандартний розмір».)

Платний працівник може таємно доручити ботові навчати вашого бота

Замість покладатися на добру волю й співпрацю незнайомців, можна платити людям за краудсорсинг. Для цього існують сервіси на кшталт Amazon Mechanical Turk: дослідник створює завдання (відповідати на запитання про зображення, грати роль працівника служби підтримки або натискати на жирафів), а віддалені виконавці отримують за це гроші. Іронічно, що стратегія може обернутися проти вас, якщо хтось візьме роботу й потайки доручить її ботові — зазвичай той виконає все жахливо. Тому користувачі платних сервісів додають прості перевірки: чи читає запитання людина, а ще краще — уважна людина, яка не відповідає навмання.[2] Інакше кажучи, їм доводиться включити до завдання тест Тюрінга, щоб переконатися, що вони випадково не найняли бота навчати власного бота.

Один приклад можна розмножити дзеркаленням і заміною слів

Інший спосіб витиснути більше з малого набору — трохи змінювати дані, перетворюючи один фрагмент на багато дещо відмінних. Це називається **розширенням даних**. Наприклад, найпростіше перетворити одне зображення на два, зробивши дзеркальну копію. Також можна вирізати частини або трохи змінити текстуру.

Розширення застосовують і до тексту, хоча рідко. Щоб перетворити кілька фраз на багато, можна замінювати їх частини словами зі схожим значенням.

> Табун коней їсть смачний торт.
> Група коней жує чудовий десерт.

> Кілька коней насолоджуються пудингом.
> Коні споживають їстівні продукти.
> Копитні пожирають кондитерське підношення.

Одне зображення показують сотні разів, щоб отримати мільйони діалогів

Автоматична генерація таких варіантів, щоправда, може породжувати дивні й малоймовірні речення. Значно частіше програмісти, які краудсорсять текст, просто дають те саме завдання багатьом людям, отримуючи безліч трохи різних відповідей з однаковим змістом. Одна команда, наприклад, створила Visual Chatbot, здатний відповідати на запитання про зображення. Вони попросили одних краудсорсерів ставити запитання, а інших — відповідати й так отримали навчальний набір із 364 мільйонів пар «запитання—відповідь». За моїми підрахунками, кожне зображення в середньому показували триста разів, тому набір містить багато майже однаково сформульованих відповідей:[3]

> ні, лише 2 жирафи
> ні, тільки 2 жирафа
> їх двоє, це не одинокий жираф, дитинча й 1 дорослий
> ні, у вольєрі лише 2 жирафи
> ні, я бачу тільки 2 жирафів
> ні, лише 2 милі жирафи
> ні, тільки 2 жирафи
> ага, тільки 2 жирафи
> ага, лише 2 жирафа
> лише 2 жирафи

Як видно з відповідей нижче, деякі учасники ставилися до серйозності проєкту відданіше за інших:

> так, я б точно зустрівся із цим жирафом
> високий жираф, мабуть, шкодує про батьківство
> птах витріщився на жирафа й запитує про крадіжку листя

Після десяти запитань про жирафа люди переходять до квантової фізики

Інший наслідок організації експерименту полягав у тому, що кожна людина мала поставити десять запитань про кожне зображення. Зрештою людям бракує того, про що ще запитати щодо жирафів, тому питання ставали трохи химерними. Серед людських запитань були такі:

> чи схоже, що жираф розуміє квантову фізику й теорію струн
> чи здається жираф достатньо щасливим, щоб зіграти в улюбленому фільмі DreamWorks

> чи схоже, що жираф з'їв людей до того, як зробили фотографію
> чи чекає жираф, поки вийдуть інші його плямисті чотириногі володарі й поневолять людство
> за шкалою від Бібера до Ґендальфа, наскільки це епічно
> ви б сказали, що це гангстерські зебри
> це схоже на елітного коня
> яка пісня жирафа
> скільки приблизно дюймів завдовжки ведмеді
> будь ласка, уважніше до завдання, ви надто довго починаєте друкувати після мого запитання, я не люблю чекати так довго, а вам подобається

Люди виробляють із наборами даних дивні речі.

І це приводить нас до наступного: недостатньо просто мати дуже багато даних. Якщо в наборі є проблеми, алгоритм у найкращому разі змарнує час, а в найгіршому — навчиться не того.

БЕЗЛАДНІ ДАНІ

Людські помилки в розмітці стають помилками безпілотного автомобіля

В інтерв'ю *The Verge* 2018 року технічний керівник напряму ШІ в Google Вінсент Вангук розповів про навчання безпілотних автомобілів. Коли дослідники виявили, що алгоритм погано позначає пішоходів, машини та інші перешкоди, вони повернулися до вхідних даних і з'ясували: більшість помилок походить від неправильної людської розмітки навчального набору.[4]

Рецепт наказує нарізати борошно кубиками й розмастити масло в холодильнику

Я теж не раз таке бачила. Одним із перших моїх проєктів була нейромережа для створення рецептів. Вона помилялася — *дуже* часто. Наприклад, наказувала кухареві:

> Змішайте мед, рідку пальцеву воду, сіль і 3 столові ложки оливкової олії.
> Наріжте борошно кубиками завтовшки ¼ дюйма.
> Розмастіть масло в холодильнику.
> Киньте одну змащену каструлю.
> Вийміть частину сковороди.

Вона просила такі інгредієнти:

> ½ склянки розривної олії
> 1 листок лекції, розморожений
> 6 квадратиків французького крему для підрум'янення
> 1 склянка італійської цільної крамбатчі

Вона безперечно насилу справлялася з масштабом і складністю створення рецептів. Її пам’яті та розумових здібностей бракувало для такої широкої задачі. Але, як з’ясувалося, деякі помилки були зовсім не її провиною. Вихідний набір містив рецепти, які якась програма колись автоматично перетворила з іншого формату, і конвертація не завжди проходила гладко.

Мережа чесно вивчає помилки, що вже були в рецептах

Один із рецептів мережі вимагав:

> 1 полуниці

— фразу, яку вона вивчила з вхідного набору. В одному рецепті вислів «2 ½ склянки нарізаної й підсолодженої свіжої полуниці», очевидно, автоматично розділився так:

> 2 ½ склянки нарізаної й підсолодженої свіжої
> 1 полуниці

Нейромережа іноді просила й нарізане борошно, але, схоже, навчилася цього на таких помилках у вихідному наборі:

> ⅔ склянки нарізаного в борошні
> 1 горіхів

Через подібні помилки нейромережа вивчила й такі інгредієнти:

> 1 (необов’язково) цукор, натертий
> 1 сіль і перець
> 1 локшини
> 1 вгору

ДАНІ, ЩО МАРНУЮТЬ ЧАС

Половина мозку рецептурної мережі йде на вигадані ISBN та примітки

Іноді проблеми в наборі не так змушували мережу помилятися, як марнували її час. Погляньте на цей створений нею рецепт:

> **Добрі понезені заправкові пустелі**
> **—ПОСИПКА—**
> 4 склянки холодної води або дріжджового м’яса
> ½ склянки масла
> ¼ чайної ложки гвоздики
> ½ склянки рослинної олії
> 1 склянка натертого білого рису
> 1 гілочки петрушки
> Готуйте цибулю в олії, борошні, фініках і солі разом через обидві тарілки.

> Покладіть соус до кожного підготовленого покритого Бройлера (2 по 10 дюймів боком догори), щоб зменшити жир, і додайте крохмаль дерев'яною зубочисткою гарячий, щоб був нижче, розтопіть курку. Прикрасьте кокосом і тертим сиром.
> Джерело: IObass Cindypissong (у Whett Quesssie. Etracklitts 6), Dallas Viewnard, Brick-Nut Markets, Fat. надіслано Fluffiting/sizevory, 1906. ISBN 0-952716-0-3015
> NUBTET 10, 1972mcTbofd-в руках, різдвяне вугілля Helb & Mochia Grunnignias: Stanter Becaused Off Matter, Dianonarddit Hht
> 5.1.85 калорій CaluAmis
> Джерело: Шоколадний пиріг, січень 584
> Вихід: 2 порції

Крім назви, категорії,* інгредієнтів та інструкцій, нейромережа половину часу створювала примітки — від джерела й харчової цінності до номера ISBN. Вона не лише марнувала час і розумові ресурси (скільки ж треба було витратити, щоб навчитися форматувати ISBN?), а й страшенно плуталася. Чому одні рецепти мають ISBN, а інші ні? Чому джерелом іноді є люди, а іноді книжки чи журнали? У навчальних даних усе це трапляється фактично випадково, тож мережа не має шансів з'ясувати приховану закономірність.

> **Местоу південносолодке з норками та начинкою у водяній свинині, барбекю**
> 3 пакунки лососевих кульок
> 1 морська сіль і перець
> 120 мм помідорів і знежиреного молока
> 2 склянки легкої сметани
> 1 склянка сухого білого вина
> 1 сіль
> 1 перець
> 1 банка 13 унцій яєць; розділених
> З'єднайте сметану із сарчболом, щоб обережно покрити м'ясисте, потім посійте й дайте йому податися (обережно ще одну ніч) (водоміґи коричного хліба, загорніть їх і покладіть, може бути готове хересом) у центр каструлі, постійно помішуючи до майже цілковитої гладкості, близько 4 хвилин. Щедро вмішайте воду, сіль, лимонний сік і картопляне пюре.
> Готуйте в маслі. Подавайте негайно. Ретельно наріжте рибу на чашку, решту 1 склянки нарізаного гороху, щоб забрати це з гриля ще на хвилину частково під і охолодженим. Воно не має зламане робить якусь-ніктивну іншу товщину. На печиво, щоб зробити полуниці.
> з Кухні Кривого, Один. Екстичний Посібник Шефа з Вермеле до приправного, це Озера ДУБ:
> **** Від Bon Meshing, 96 1994. MG (8Fs4.TE, Від: Hoycoomow Koghran*.Lavie: 676 (WR/12-92-1966) entral. Пірнайте ними, Tiftigs: ==1
> Поділився: Dandy Fistary

> Вихід: 10 порцій

92 тисячі заголовків BuzzFeed змушують мережу одночасно вчити кілька мов

В іншому експерименті я навчала нейромережу вигадувати назви статей-списків BuzzFeed. Перший раунд вийшов невдалим. Ось кілька створених заголовків:

> 11 Videos Unges Annoying Too Real Week
> 29 choses qui aphole donnar desdade
> 17 Things You Aren’t Perfectly And Beautiful
> 11 choses qui en la persona de perdizar como
> 11 en 2015 fotos que des zum Endu a ter de viven beementer aterre Buden
> 15 GIFs
> 14 Reasons Why Your Don’t Beauty School Things Your Time
> 11 fotos qui prouitamente tu pasan sie de como amigos para
> 18 Photos That Make Book Will Make You Should Bengulta Are In 2014
> 17 Reasons We Astroas Admiticational Tryihnall In Nin Life

Половина статей була не англійською, а якоюсь дивною сумішшю французької, іспанської, німецької та ще кількох мов. Я повернулася до набору й справді

побачила: хоча він містив вражаючі дев’яносто дві тисячі заголовків, половина була написана не англійською. Нейромережа витрачала половину часу на англійську, а половину — на спроби одночасно вивчити й розділити кілька інших мов. Коли я прибрала зайві, англійські результати теж поліпшилися:

> 17 разів найбільше дуп
> 43 цитати, що гарантовано негайно зроблять вас русалкою
> 31 фотографія костюма волосся черепашок-ніндзя
> 18 таємниць, яких вам не розкажуть сніговики
> 15 емо-футбольних фанатів діляться своїми способами
> 27 різдвяних прикрас, які знає кожен двадцятирічний студент
> 12 серйозних творчих способів розмістити курячі місця в Сіднеї
> 25 невдалих печивних вистав з усього світу
> 21 фотографія їжі, від якої ви скривитеся й скажете: «о, мені я сумно?»
> 10 спогадів, які зроблять вас здоровими у 2015 році
> 24 рази, коли Австралія була абсолютно найгіршою
> 23 меми про те, як бути смішним, що смішні, але також сміються з
> 18 смачних ласощів із беконом, що зроблять клоунів неймовірно щасливими
> 29 речей, які можна зробити з чаєм на Гелловін
> 7 пирогів
> 32 ознаки волохатого тата

Алгоритми машинного навчання не знають контексту наших задач, тому не розуміють, що важливе, а що слід ігнорувати. Нейромережа заголовків BuzzFeed не знала про існування різних мов і про те, що ми хотіли лише англійських результатів; з її погляду всі закономірності були однаково важливими. Зосередження на зайвих відомостях дуже поширене й серед алгоритмів генерації та розпізнавання зображень.

Котячі меми змушують GAN малювати псевдотекст і Grumpy Cat

У 2018 році команда Nvidia навчила GAN створювати різні зображення, зокрема котів.[5] Вони помітили, що деяких котів супроводжували квадратні позначки, схожі на текст. Очевидно, у навчальних даних були котячі меми, й алгоритм сумлінно витрачав час, з’ясовуючи, як створювати мемний текст. У 2019 році інша команда на тому самому наборі навчила StyleGAN, яка теж додавала до котів псевдотекст. Вона також присвятила чимало часу зображенню одного незвичайного, але знаменитого в інтернеті кота — Grumpy Cat.[6]

BigGAN плутає футбольний м’яч із ногою, а мікрофон — із людиною

Інші генератори зображень плутаються так само. У 2018 році команда Google навчила BigGAN, яка вражаюче добре створювала різноманітні картинки.

Особливо вдало виходили собаки (у наборі було *дуже* багато прикладів) і пейзажі (мережі чудово давалися текстури). Але навчальні зображення іноді її заплутували. На картинках за запитом «футбольний м'яч» часом з'являлася м'ясиста грудка — мабуть, спроба зобразити людську стопу — або навіть цілий воротар; а «мікрофони» часто були людьми без помітного мікрофона. Навчальні фото не показували предмет ізольовано: там були люди й тло, про які нейромережа теж намагалася дізнатися. На відміну від людини, BigGAN не могла відрізнити оточення об'єкта від самого об'єкта — пам'ятаєте плутанину між пейзажем і вівцею з розділу 1? Як StyleGAN не впоралася з усіма видами котячих фото, так BigGAN боролася з набором, що ненавмисно зробив її задачу надто широкою.

Очищення набору й обведення предметів допомагають алгоритмові

Якщо набір безладний, один із головних способів поліпшити результат — витратити час на очищення. Програмісти можуть піти далі й використати власне знання даних: наприклад, відсіяти фотографії футбольних м'ячів, де є щось іще — воротарі, пейзажі та сітки. Для алгоритмів розпізнавання люди також можуть малювати рамки чи контури навколо різних предметів, вручну відокремлюючи потрібний об'єкт від тих, із якими він часто трапляється.

Та навіть чисті дані нерідко містять проблеми.

ЦЕ СПРАВЖНЄ ЖИТТЯ?

Раніше в цьому розділі я згадувала: навіть порівняно чисті дані без великої кількості зайвого мотлоху можуть змусити ШІ гепнутися обличчям, якщо вони не відображають реального світу.

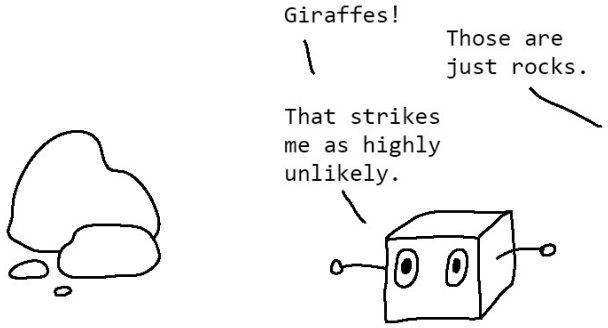
Візьмімо, наприклад, жирафів.

Через фотографічні звички людей ШІ бачить жирафів у кожному пейзажі

У спільноті дослідників і шанувальників ШІ має репутацію того, хто бачить жирафів усюди. Дайте йому випадкову фотографію нецікавого пейзажу — ставка чи дерев — і він, імовірно, повідомить про наявність жирафів. Ефект настільки поширений, що експертка з інтернет-безпеки Мелісса Елліотт запропонувала термін *giraffing* («жирафлення») для явища, коли ШІ надмірно часто повідомляє про порівняно рідкісні речі.[7]

Причина — у навчальних даних. Хоча жирафи трапляються нечасто, люди набагато охочіше фотографують жирафа («Гей, дивіться, жираф!»), ніж випадковий нудний шматок пейзажу. Великі безплатні набори зображень, на яких так багато дослідників навчають алгоритми, містять безліч різних тварин, але мало — якщо взагалі хоч трохи — фотографій звичайної землі чи звичайних дерев. ШІ, який вивчає цей набір, засвоює, що жирафи поширеніші за порожні поля, і відповідно коригує передбачення.

Я перевірила це на Visual Chatbot: хоч би які нудні фотографії показувала, бот був переконаний, що потрапив на найкраще сафарі у своєму житті.



Непредставлені у Wikipedia науковиці стають невидимими й для ШІ

«Зажирафлений» ШІ чудово відповідає побаченим даним, але погано — реальному світові. У навчальних наборах надмірно або недостатньо представлені не лише тварини й земля, а всілякі речі. Наприклад, люди зауважували, що жінки-науковиці у *Wikipedia* представлені набагато гірше за чоловіків із подібними досягненнями. Донна Стрікленд, лауреатка Нобелівської премії з фізики 2018 року, не мала статті у *Wikipedia* до самої перемоги; лише на початку того року чернетку про неї відхилили, бо редактор вирішив, що вона недостатньо відома.[8] ШІ, навчений на статтях *Wikipedia*, може вирішити, що визначних жінок-науковиць майже немає.

ІНШІ ДИВАЦТВА НАБОРІВ ДАНИХ

Нісенітниця сомалійською раптом перетворюється на біблійні пророцтва

Особливості конкретного набору іноді дивовижно проявляються в навченій моделі. У 2018 році користувачі Google Translate помітили: коли вони просили перекласти з деяких мов на англійську повторювані безглузді склади, результат був дивно зв’язним — і дивно біблійним.[9] Джон Крістіан із *Motherboard* дослідив це й виявив, наприклад, що

> «ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag ag»

перекладалося із сомалійської англійською як

> «Унаслідок цього загальна кількість членів племені синів Ґершона становила сто п’ятдесят тисяч»

тоді як

> «ag ag ag ag ag ag ag ag ag ag»

перекладалося із сомалійської англійською як

> «А довжина його з одного краю становила сто ліктів»

Після того як *Motherboard* звернувся до Google, дивні переклади зникли. Але лишилося питання: чому це взагалі сталося? Редактори опитали експертів із машинного перекладу, які припустили, що причина — використання машинного навчання. Алгоритм навчається перекладати слова й фрази, розглядаючи приклади, уже перекладені людьми. Він засвоює, які фрази відповідають одна одній і в якому контексті. Завдяки цьому зазвичай добре створює природні переклади, навіть ідіом і сленгу. Алгоритм Google став одним із перших масштабних комерційних застосувань машинного навчання й привернув увагу світу у 2010 році, коли майже за одну ніч різко поліпшив сервіс перекладу. З розділу 2 ми знаємо: алгоритм працює найкраще, коли має багато прикладів. Експерти припустили, що для деяких мов Google Translate мав мало паралельних перекладів, але серед них майже напевно була Біблія, адже її переклали величезною кількістю мов. Коли алгоритм не був певний у перекладі, він міг за замовчуванням відтворювати шматки навчальних даних — звідси дивні релігійні уривки.

Зміна пробілів змушує перекладач вигадувати форум та електронну адресу

Коли я перевірила це наприкінці 2018 року, біблійні уривки вже зникли, але Google Translate усе ще дивно поводився з повторюваними й безглуздими складами.*

Наприклад, я змінювала пробіли в англійському реченні, а потім просила перекласти отриману нісенітницю з маорійської на англійську. Ось кілька результатів:

> ih ave noi dea wha tthi ssen tenc eis sayi ng →
> Ваша електронна адреса — одна з найважливіших функцій цього форуму
> ih ave noi dea wha tthi ssen tenc eis sayi ngat all →
> Це один із найкращих способів придбати одну або кілька таких речей
> ih ave noi dea wha tthi ssen tenc eis sayi ngat all ple aseh elp →

> Крім того, ви зможете дізнатися більше про запити

Це явище дивне й веселе, але має й серйозний бік. Багато закритих нейромереж навчаються на інформації клієнтів, частина якої може бути надзвичайно приватною та конфіденційною. Якщо навчену модель можна розпитати так, щоб вона розкрила відомості з навчальних даних, це створює величезний ризик безпеки.

Нейромережа може видати номери карток і соціального страхування

У 2017 році дослідники Google Brain показали, що стандартний алгоритм мовного перекладу здатен запам’ятати короткі послідовності чисел — номери кредитних карток або соціального страхування, — навіть якщо вони трапляються лише чотири рази в наборі зі ста тисяч англо-в’єтнамських пар речень.[10] Не маючи доступу ані до навчальних даних ШІ, ані до його внутрішньої будови, дослідники виявили, що він упевненіший у перекладі, якщо це точна пара речень, яку бачив під час навчання. Змінюючи цифри в тестовому реченні «Мій номер соціального страхування — XXX-XX-XXXX», вони могли визначити, які номери ШІ зустрічав раніше. Вони навчили РНМ на понад ста тисячах електронних листів із чутливою інформацією працівників, яку уряд США зібрав у межах розслідування діяльності корпорації Enron (так, тієї самої Enron), і змогли витягнути з передбачень кілька номерів соціального страхування та кредитних карток. Мережа запам’ятала дані так, що їх міг відновити будь-який користувач — навіть без доступу до вихідного набору. Цю проблему називають **ненавмисним запам’ятовуванням**. Їй можна запобігти належними заходами безпеки — або від початку не включати чутливі дані до навчального набору.

ВІДСУТНІ ДАНІ

Ось іще один спосіб саботувати ШІ: не дати йому всієї необхідної інформації.

Список котячих імен не містить слів, якими кота називати не варто

Люди використовують *дуже* багато відомостей навіть для найпростіших рішень. Припустімо, ми вибираємо ім’я котові. Можемо пригадати безліч знайомих котів і приблизно зрозуміти, як має звучати котяче ім’я. Нейромережа теж здатна переглянути довгий список наявних імен, виявити поширені комбінації літер і навіть найчастіші слова. Але вона не знає слів, яких *немає* в цьому списку. Люди розуміють, яких слів уникати; ШІ — ні. Тому список котячих імен, створений рекурентною нейромережею, міститиме таке:

> Hurler — Блювач
> Hurker — Ригун
> Jexley Pickle — Джекслі Огірок
> Sofa — Диван
> Trickles — Цівочки
> Clotter — Згустник
> Moan — Стогін
> Toot — Пук
> Pissy — Зассанець
> Retchion — Блювотіон
> Scabbys — Коростики
> Mr Tinkles — Пан Дзюрчик

За звучанням і довжиною вони чудово вписуються серед решти котячих імен. Із цією частиною ШІ впорався добре. Але випадково вибрав кілька справді дуже дивних слів.

Коли дивність є метою, обмежене знання світу стає перевагою

Іноді дивність — саме те, що треба, і тут нейронні мережі сяють. Працюючи з літерами та звуками, а не зі значеннями й культурними посиланнями, вони здатні створювати комбінації, що навряд чи спали б на думку людині. Пам’ятаєте зібраний краудсорсингом список гелловінських костюмів? Ось що вигадала РНМ, коли я попросила її наслідувати їх:

> Пташиний чарівник
> Диско-монстр
> Мім Похмурого Женця
> Спартанський Ґендальф
> Кінь-міль
> Акула Зоряного флоту
> Коробка в масці
> Панда-молюск
> Акула-корова
> Шкільний автобус-зомбі
> Опудало Снейпа
> Професор Панда
> Полунична акула
> Король жука-какашки
> Невдалий стімпанковий павук
> Пані Сміття
> Робот пані Фрізл
> Селерово-синій Франкенштейн
> Дракон Свободи
> Принцеса-акула

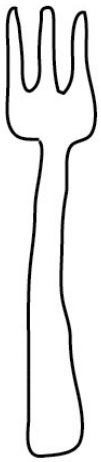
> Штани-кексик
> Привид Огірка
> Наречена вампірського кабана
> Статуя піци
> Гарбузовий Пікар

Текстові РНМ створюють нісенітниці, бо їхній світ, по суті, *і є* нісенітницею. Якщо конкретних прикладів немає в наборі, нейромережа не розуміє, чому «Шкільний автобус-зомбі» малоймовірний, а «Чарівний шкільний автобус» має сенс, або чому «Привид Огірка» — менш імовірний вибір, ніж «Привид минулого Різдва». На Гелловін це навіть корисно: частина веселощів — бути єдиною людиною на вечірці в костюмі «Нареченої вампірського кабана».

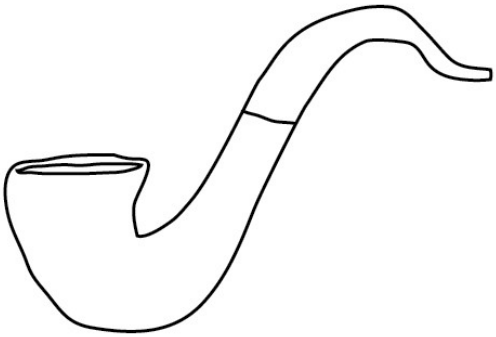
Через обмежене й вузьке знання світу ШІ може спіткнутися навіть на доволі буденних речах. Наше «буденне» все одно надзвичайно широке, й створити ШІ, готовий до всього, дуже складно.

Azure бачить у лінійному малюнку текстуру, якої там немає

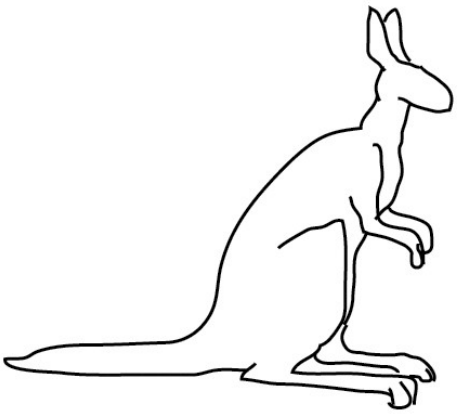
Творці алгоритму розпізнавання Microsoft Azure — того самого ШІ, який бачив овець у кожному полі, — задумали його так, щоб він правильно описував будь-який завантажений файл: фотографію, картину або навіть лінійний малюнок. Тож я дала йому кілька ескізів.



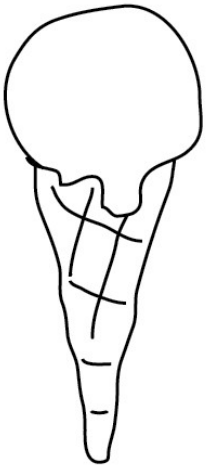
a close up of a device



a close up of a lamp



a drawing of a map



a close up of a basketball hoop

Малюю я не чудово, але й не *настільки* погано. Це просто алгоритм, який намагається робити забагато. Розпізнавати будь-який файл зображення — майже пряма протилежність вузьким задачам, де ШІ найсильніший. Більшість зображень, які Azure бачив під час навчання, були фотографіями, тому він дуже покладається на текстури: це хутро? трава? У моїх лінійних малюнках текстур немає, й алгоритмові бракує досвіду, щоб їх зрозуміти. (Утім, Azure

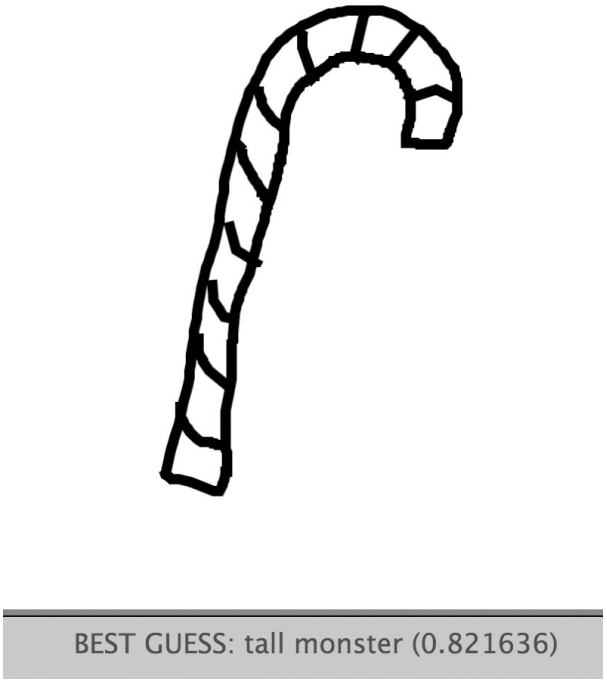
впорався краще за багато інших розпізнавачів, які будь-який лінійний малюнок називають різновидом «UNK» — невідомого.) Дослідники навчають алгоритми на мультфільмах, малюнках і фотографіях із сильно зміненими текстурами: якщо ШІ розуміє побачене так само добре, як людина, він має розібратися й у мультфільмах.

Quick Draw знає лише 345 речей — зате мільйони їхніх ескізів

Існує алгоритм, спеціально створений для простих ескізів. Дослідники Google навчили Quick Draw на мільйонах малюнків, попросивши людей грати з комп’ютером у щось на кшталт «Крокодила». Тому алгоритм розпізнає ескізи понад трьохсот різних об’єктів попри дуже різні художні здібності людей. Ось невелика добірка навчальних ескізів *кенгуру*:[11]

Quick Draw одразу впізнав мого кенгуру.[12] Він також упізнав виделку й ріжок морозива. Труба завдала труднощів, бо не входила до 345 знайомих йому предметів. Алгоритм вирішив, що це лебідь або садовий шланг.

Насправді, знаючи лише ці 345 речей, Quick Draw відповідав на багато моїх ескізів цілковитою дивовижею.



Автодоповнення правильно передбачає похорон бабусі — і саме тому помиляється

Усе це чудово, якщо ви, як і я, зробили дивність своєю метою. Але неповна картина світу створює проблеми в деяких застосуваннях — наприклад, в автодоповненні. З розділу 3 ми знаємо, що передбачення тексту у смартфонах зазвичай працює на різновиді машинного навчання під назвою ланцюг Маркова. Та компаніям важко не дозволити ШІ безтурботно пропонувати гнітючі або образливі варіанти. Як Даан ван Есх, керівник проєкту застосунку автовиправлення Android GBoard, розповів інтернет-лінгвістці Ґретхен Маккаллок: «Певний час, коли ви друкували “Я йду до бабусі на…”, GBoard фактично пропонував “похорон”. Саме по собі це не *неправильно*. Можливо,

таке трапляється частіше, ніж “бабусину рейв-вечірку”. Але водночас про це не хочеться згадувати. Тож краще бути трохи обережним». ШІ не знає, що цілком точне передбачення все одно не є правильною відповіддю, тому інженерам доводиться втручатися й навчати його не пропонувати це слово.[13]

ТУТ ЧОТИРИ ЖИРАФИ

Якщо люди питають лише про видимі речі, ШІ навчається відповідати «так»

Visual Chatbot, ШІ для відповідей на запитання про зображення, має чимало цікавих дивацтв, пов’язаних із даними. Творці навчали його на краудсорсинговому наборі запитань і відповідей до картинок. Тепер ми знаємо, що упередження даних перекошує відповіді ШІ, тому програмісти організували збір так, щоб уникнути кількох відомих упереджень. Одним було **візуальне праймування**: дивлячись на зображення, люди схильні ставити запитання, відповідь на які — «так». Вони майже ніколи не питають: «Ви бачите тигра?» — коли тигрів немає. Тому ШІ, навчений на таких даних, засвоїть, що відповідь на більшість запитань — «так». В одному випадку алгоритм із упередженого набору досягав 87-відсоткової точності, відповідаючи «так» на кожне запитання, що починалося словами «Ви бачите…». Якщо це знайоме, пригадайте дисбаланс класів із розділу 3: велика партія переважно жахливих сендвічів привела ШІ до висновку «Люди ненавидять усі сендвічі».

Щоб уникнути візуального праймування, під час краудсорсингового збору програмісти ховали зображення від людей, які ставили запитання. Змушуючи їх формулювати загальні запитання з відповіддю «так» або «ні», придатні для будь-якої картинки, вони отримали приблизний баланс позитивних і негативних відповідей.[14] Але й цього не вистачило, щоб усунути проблеми.

Запитання про кількість жирафів майже ніколи не має відповіді «нуль»

Одне з найкумедніших дивацтв набору: хоч би що було на картинці, на запитання «Скільки тут жирафів?» Visual Chatbot майже завжди відповідає, що є принаймні один. До цього моменту він може непогано описувати людей на нараді чи серферів на хвилі. Але потім його питають про жирафів — і практично незалежно від зображення бот повідомляє про одного, чотирьох або навіть «надто багато, щоб порахувати».

Джерело проблеми? Під час збору даних люди рідко запитували «Скільки тут жирафів?», коли відповідь була «нуль». А навіщо? У звичайній розмові ніхто не влаштовує іншому вікторину про кількість жирафів, коли обидва знають, що їх немає. Отже, Visual Chatbot підготували до нормальної людської розмови, обмеженої правилами ввічливості, але не до дивних людей, які питають про випадкових жирафів.

Блакитне яблуко для бота все одно червоне, а помаранчева вівця — біла

Через навчання на звичайних розмовах між звичайними людьми ШІ геть не готовий і до інших форм дивини. Покажіть Visual Chatbot блакитне яблуко, й на запитання «Якого кольору яблуко?» він відповість «червоне», «жовте» або назве інший нормальний яблучний колір. Замість складного завдання розпізнавати колір предмета бот засвоїв, що відповідь на «Якого кольору яблуко?» майже завжди — «червоне». Так само, побачивши вівцю, пофарбовану яскраво-синім чи помаранчевим, на запитання про колір він повідомить стандартний овечий варіант: «чорно-біла» або «біло-коричнева».

Коли бот не розуміє BB-8, він упевнено оголошує його собакою

Насправді Visual Chatbot має небагато засобів для вираження невпевненості. У навчальних даних люди зазвичай розуміли, що відбувається на картинці, навіть якщо на деякі запитання — наприклад, «Що написано на знаку?» — не можна було відповісти, бо знак затуляли. На запитання «Якого кольору X?» бот навчився казати: «Не можу визначити; зображення чорно-біле», — навіть коли воно очевидно кольорове. На «Якого кольору її капелюх?» відповість: «Не можу визначити; я не бачу її ніг». Він вигадує правдоподібні виправдання для нерозуміння, але в цілком неправильному контексті. Одного майже ніколи не робить: не виражає загальної розгубленості, бо люди, на яких він навчався, не були розгублені. Покажіть йому BB-8, кулястого робота із *«Зоряних війн»*, — Visual Chatbot оголосить, що це собака, і далі відповідатиме так, ніби бачить собаку. Інакше кажучи, він блефує.

Обмежена модель світу мусить вгадувати серед безмежної людської дивини

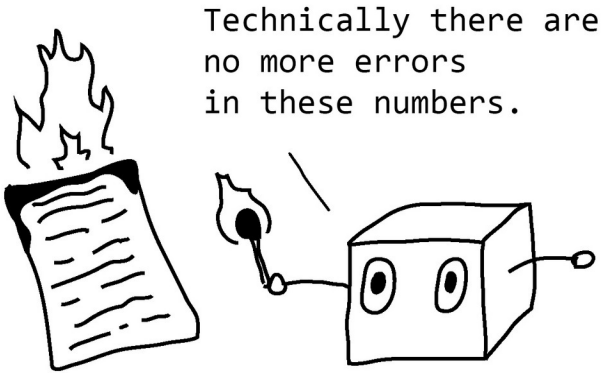
Під час навчання ШІ бачить лише обмежену кількість речей, і це проблема для застосувань на кшталт безпілотних автомобілів, які мають зустрічатися з безмежною дивиною людського світу й вирішувати, як діяти. Як я згадувала в частині про безпілотні автомобілі у розділі 2, керування на справжніх дорогах — дуже широка задача. Як і взаємодія з усім, що людина може сказати чи

намалювати. У підсумку ШІ робить найкраще припущення на основі обмеженої моделі світу й іноді помиляється смішно — або трагічно.

У наступному розділі ми розглянемо ШІ, які чудово розв’язали поставлені задачі, — от тільки ми випадково поставили їм не ті задачі.

РОЗДІЛ 5. Чого ви насправді просите?

Чотири способи виконати завдання зовсім не так



Якось я намагався написати нейронну мережу, яка максимізувала б прибуток від ставок на кінські перегони. Вона визначила, що найкраща стратегія — *барабанний дріб* — не робити жодної ставки.

—@citizen_of_now[1]

Я намагався еволюційно навчити робота не врізатися в стіни:
1) Він еволюціонував так, щоб не рухатися, а отже, не врізався в стіни
2) Додав винагороду за рух: він закрутився
3) Додав винагороду за рух убік: почав їздити маленькими колами
4) і так далі.
Підсумкова назва книжки: «Як еволюційно вивести програміста»

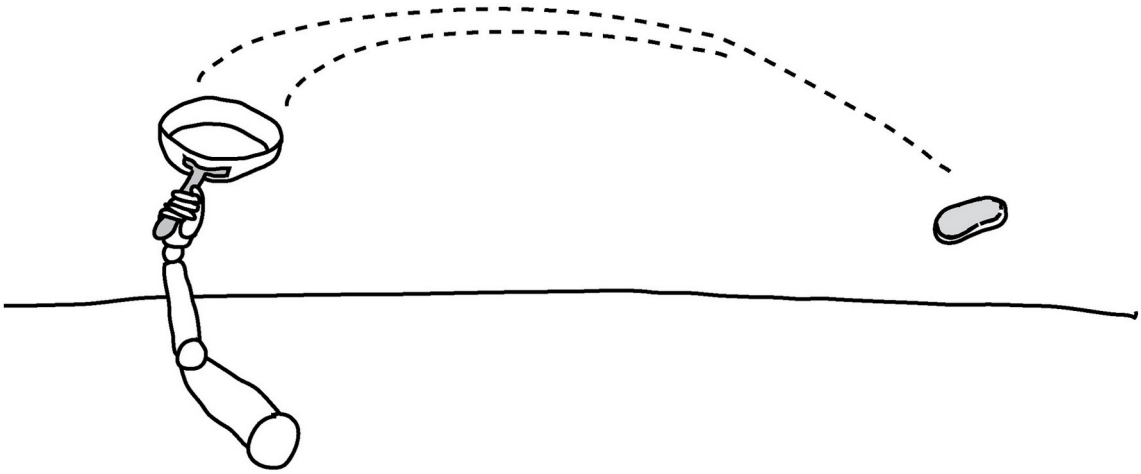
—@DougBlank[2]

Я під’єднав нейронну мережу до свого Roomba. Я хотів, щоб він навчився пересуватися, ні в що не врізаючись, тож налаштував систему винагород, яка заохочувала швидкість і карала спрацювання датчиків бампера. Він навчився їздити задом, бо ззаду бамперів немає.

—@smingleigh[3]

Моя мета — навчити роботизовану руку готувати млинці. Для першого випробування [я спробувала] змусити руку підкинути млинець так, щоб він упав на тарілку… Перша система винагород була простою: за кожен кадр сеансу нараховувалася невелика винагорода, а сеанс завершувався, коли млинець падав на підлогу. Я думала, це спонукатиме алгоритм якомога довше тримати млинець на пательні. Насправді ж він намагався жбурнути млинець якнайдалі, максимально збільшуючи час його перебування в повітрі… Рахунок — МлинцеБот: 1, я: 0.

—Крістін Баррон[4]



Чому ШІ розв’язує не ту задачу

Як ми вже бачили, існує багато способів ненароком зіпсувати роботу ШІ, надавши йому хибні або недостатні дані. Але є й інший різновид невдачі ШІ: ми виявляємо, що він успішно зробив те, про що ми просили, — просто просили ми не того, чого насправді хотіли.

Чому ШІ так схильні розв’язувати не ту задачу?

1. Вони виробляють власні способи розв’язання задачі, а не покладаються на покрокові вказівки програміста.
2. Їм бракує контекстних знань, щоб зрозуміти, коли їхні рішення — не ті, яким віддали б перевагу люди.

Хоча саме ШІ виконує роботу з пошуку способу розв’язання задачі, програмістові все одно треба переконатися, що ШІ справді розв’язав правильну задачу. Зазвичай для цього доводиться багато працювати над двома речами:

3. Визначити мету досить чітко, щоб обмежити ШІ корисними відповідями.
4. Перевірити, чи не примудрився ШІ, попри все, вигадати некорисне рішення.

Сформулювати мету так, щоб ШІ випадково не витлумачив її неправильно, надзвичайно складно. Особливо якщо його хибна версія завдання простіша за те, чого ви від нього хочете.

Проблема в тому, що, як ми бачили впродовж усієї цієї книжки, ШІ розуміють свої завдання далеко не настільки добре, щоб брати до уваги контекст, етику чи основи біології. ШІ може класифікувати зображення легень як здорові або

хворі, так і не зрозумівши, як працюють легені, якого вони розміру чи навіть що вони містяться всередині людини — хоч би що таке людина. У нього немає здорового глузду, і він не знає, коли слід попросити уточнення. Дайте йому мету — дані для наслідування або **функцію винагороди**, яку треба максимізувати (наприклад, пройдену відстань чи очки, зібрані у відеогрі), — і він її виконає, незалежно від того, чи розв'язав він вашу задачу насправді.

Програмісти, які працюють зі ШІ, навчилися ставитися до цього філософськи.

ШІ-демон навмисно шукає найлегшу лазівку

«Я почав уявляти [ШІ] демоном, який навмисно хибно тлумачить вашу винагороду й активно шукає найлегший можливий локальний оптимум. Звучить трохи безглуздо, але я виявив, що такий спосіб мислення насправді продуктивний», — пише Алекс Ірпан, дослідник ШІ в Google.[5]

Робопси, які не хочуть ходити

Спроби іншого роздратованого програміста навчити віртуальних роботів-псів ходити породили псів, які посмикуваннями пересувалися по землі, виконували дивні віджимання зі схрещеними задніми лапами й навіть зламували фізику симуляції, щоб зависати в повітрі.[6] Як написав у Twitter інженер Стерлінґ Кріспін:

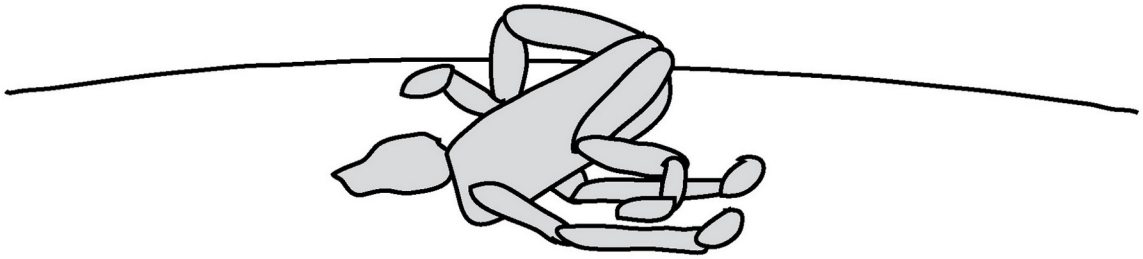
> Я думав, що просуваюся вперед… але ці ПОКИДЬКИ просто знайшли ваду у фізичній симуляції й тепер ковзають над підлогою, як останні шахраї.

Борючись зі схильністю роботів робити будь-що, *крім* ходіння, Кріспін раз у раз підправляв їхню функцію винагороди: запровадив «штраф за чечітку», щоб вони перестали швидко дріботіти на місці, і «винагороду за те, щоб, чорт забирай, торкатися землі», аби, власне, розв'язати проблему зависання. У відповідь вони почали безрезультатно соватися по землі. Тоді Кріспін запровадив винагороду за те, щоб вони тримали тулуб над землею, а коли вони почали човгати, задерши зад догори, — винагороду за рівне положення тулуба. Щоб вони перестали вперто ходити зі схрещеними задніми лапами, Кріспін винагороджував їх за те, що нижні частини лап не торкалися землі; щоб вони перестали хитатися, він запровадив *ще одну* винагороду за рівне положення тулуба — і так далі. Важко було зрозуміти, що це: доброзичливий програміст підказує робопсам, як користуватися лапами, чи змагання волі між програмістом і робопсами, які. Не. Хотіли. Ходити. (Також виникла невеличка складність, коли робопси вперше зіткнулися з чимось іншим, ніж ідеально

рівна й гладенька поверхня, яку бачили під час навчання. Потрапивши на трохи текстуровану землю, вони падали мордою вперед.)

Собаки й алгоритми вчаться не того, чого ми очікуємо

Виявляється, навчання алгоритму машинного навчання має багато спільного з дресируванням собак. Навіть якщо пес дуже хоче співпрацювати, люди можуть випадково навчити його робити не те. Наприклад, у собак настільки чудовий нюх, що вони здатні виявляти запах раку в людей. Але тим, хто навчає собак винюхувати рак, треба давати їм різних пацієнтів, інакше вони навчаться розпізнавати окремих людей, а не рак.[7] Під час Другої світової війни існував доволі моторошний радянський проєкт із навчання собак доставляти бомби до ворожих танків.[8] Виникло кілька труднощів:



5. Собак навчали діставати їжу з-під танків, але для економії пального й боєприпасів танки не рухалися й не стріляли. Собаки не знали, що робити з рухомими танками, а стрілянина їх лякала.
6. Радянські танки, на яких тренувалися собаки, пахли інакше, ніж німецькі танки, які вони мали шукати: німецькі працювали на бензині, а радянські — на дизельному пальному.

Унаслідок цього в бою собаки здебільшого уникали німецьких танків, розгублено поверталися до радянських солдатів і навіть шукали радянські танки. Радянських солдатів це зовсім не тішило, адже собаки все ще несли свої бомби.

Мовою машинного навчання це **перенавчання**: собаки були підготовлені до умов, які бачили під час тренувань, але ці умови не відповідали реальному світові. Так само робопси перенавчилися на дивній фізиці своєї симуляції, використовуючи стратегії зависання й ковзання, які ніколи не спрацювали б у реальному світі.

Є ще одна риса, якою навчання тварин схоже на навчання алгоритмів машинного навчання: руйнівний вплив хибної функції винагороди.

ЗЛАМУВАННЯ ФУНКЦІЇ ВИНАГОРОДИ

Дельфіни розмінюють сміття на рибу

Дресирувальники дельфінів з’ясували, що зручно залучати тварин до прибирання їхніх басейнів. Треба лише навчити дельфінів знаходити сміття й приносити доглядачам в обмін на рибину. Утім, це не завжди добре працює. Деякі дельфіни помічають, що обмінний курс однаковий незалежно від розміру шматка сміття, і починають приховувати сміття замість того, щоб віддавати його, відриваючи маленькі шматочки й приносячи доглядачам кожен за окрему рибину.[9]

Люди теж зламують стимули

Люди, звісно, теж зламують свої функції винагороди. У розділі 4 я згадувала: люди, які наймають інших людей створювати навчальні дані через віддалені сервіси на кшталт Amazon Mechanical Turk, інколи виявляють, що роботу натомість виконали боти. Це можна вважати випадком хибної функції винагороди: якщо платять за кількість відповідей, а не за їхню якість, то справді фінансово вигідніше створити ботів, які дадуть за вас багато відповідей, ніж самому відповісти на кілька запитань. Так само багато різновидів злочинів і шахрайства можна розглядати як зламування функції винагороди. Навіть лікарі можуть зламувати свої функції винагороди. У США рейтингові картки лікарів мають допомагати пацієнтам обирати найефективніших фахівців і уникати тих, у кого після операцій виживає менше пацієнтів, ніж у середньому. Також вони мають спонукати лікарів покращувати результати. Натомість деякі лікарі почали відмовляти пацієнтам, чиї операції будуть ризикованими, щоб не псувати свої рейтингові картки.[10]

Люди, однак, зазвичай хоч трохи розуміють, що саме функція винагороди *мала* заохочувати, навіть якщо не завжди вирішують грати за правилами. ШІ такого поняття не має. Річ не в тому, що він прагне нас здолати чи *намагається* шахраювати, — просто його віртуальний мозок приблизно завбільшки з мозок черв’яка, і за один раз він може навчитися лише одного вузького завдання. Навчіть ШІ відповідати на запитання про людську етику — і це буде все, що він уміє: він не зможе керувати автомобілем, розпізнавати обличчя чи відбирати резюме. Він навіть не зможе розпізнавати етичні дилеми в історіях і обмірковувати їх: розуміння історій — цілком інше завдання.

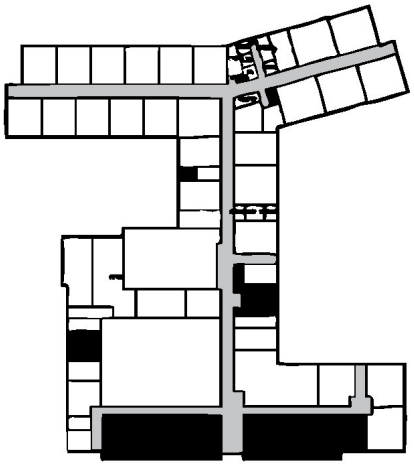
Пожежі без заторів і школи без вікон

Саме тому виникають алгоритми на кшталт навігаційного застосунку, який під час каліфорнійських лісових пожеж у грудні 2017 року спрямовував автомобілі

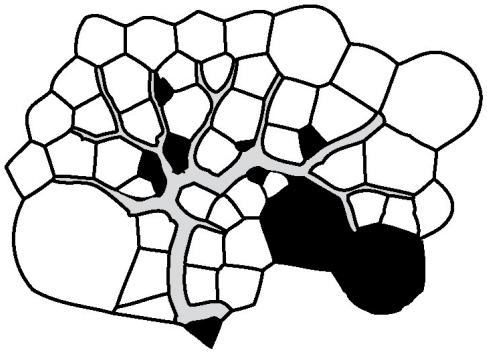
до охоплених вогнем районів. Він не намагався вбивати людей: просто побачив, що там менше заторів. Ніхто не розповів йому про пожежу.[11]

Саме тому, коли фахівець з інформатики Джоел Саймон застосував генетичний алгоритм, щоб розробити нове, ефективніше планування початкової школи, його перші проєкти мали класи без вікон, заховані глибоко в центрі лабіринту печер із круглими стінами. Ніхто не розповів алгоритмові про вікна, плани пожежної евакуації чи про те, що стіни мають бути прямими.[12]

Standard school plan



AI’s optimized school plan



Поні з іменами, яких краще уникати

Саме тому виникають і такі алгоритми, як RNN, яку я навчила вигадувати нові імена для «My Little Pony», наслідуючи список наявних імен поні. Вона знала, які поєднання літер трапляються в іменах поні, але не знала, яких поєднань краще уникати. У підсумку я отримала ось таких поні:

> Rade Slime (Сяйний Слиз)
> Blue Cuss (Синя Лайка)
> Starlich (Зоряний Ліч)
> Derdy Star (Брудна Зірка)
> Pocky Mire (Віспяне Болото)
> Raspberry Turd (Малинове Лайно)
> Parpy Stink (Пукливий Смердюк)
> Swill Brick (Помийна Цеглина)
> Colona (Колона)
> Star Sh*tter (Зоряний Срунь)

І саме тому виникають алгоритми, які вчаться, що расова й гендерна дискримінація — зручний спосіб наслідувати людей у своїх наборах даних. Вони не знають, що наслідувати упередження неправильно. Вони знають лише, що це закономірність, яка допомагає їм досягати мети. Забезпечити етику й здоровий глузд мусить програміст.

КОМП'ЮТЕРНІ ІГРИ ЗБИВАЮТЬ З ПАНТЕЛИКУ

«Перемогти» — надто нечітка мета

Популярне випробувальне завдання для ШІ — навчитися грати в комп'ютерні ігри. Ігри цікаві: на них зручно проводити демонстрації, а багато найперших комп'ютерних ігор працюють на сучасних машинах дуже швидко, тож ШІ може за прискорений час награти тисячі годин.

Та навіть найпростіші комп'ютерні ігри ШІ буває дуже важко пройти — часто тому, що йому потрібні дуже конкретні цілі. Найкращі цілі — ті, щодо яких алгоритм може одразу отримати зворотний зв'язок: чи правильно він діє. Отже, «перемогти у грі» — погана мета, а «збільшити свій рахунок» і навіть «залишатися живим якомога довше» можуть бути добрими. Проте навіть із добрими цілями алгоритми машинного навчання все одно можуть насилу розуміти поставлене завдання.

Tetris: поставити гру на вічну паузу

У 2013 році один дослідник створив алгоритм для класичних комп'ютерних ігор. Граючи в Tetris, алгоритм начебто навмання розставляв блоки, дозволяючи їм накопичитися майже до самого верху екрана. Тоді алгоритм усвідомлював, що програє, щойно з'явиться наступний блок, і тому… ставив гру на паузу назавжди.[13,14]

Насправді «поставити гру на паузу, щоб не сталося нічого поганого», «залишатися на самому початку рівня, де безпечно» чи навіть «померти наприкінці першого рівня, щоб тебе не вбив другий» — усе це стратегії, які алгоритми машинного навчання застосовуватимуть, якщо їм дозволити. Наче в ігри грають дуже буквальні малюки.

Mario не знає, що помирати погано

Якщо ШІ *не* сказати, що втрачати життя не можна, він ніяк не дізнається, що не повинен помирати. Один дослідник зумів навчити ШІ грати в Super Mario: той пройшов увесь другий рівень, а на початку третього відразу ж стрибнув у яму й загинув. Програміст дійшов висновку, що ШІ, якому прямо не наказали

берегти життя, гадки не мав, що зробив щось погане. Після смерті його повернули на початок рівня, але оскільки він і так був зовсім близько до початку, то не надто розумів, у чому проблема.[15]

Вітрильник нескінченно збирає ті самі маркери

Інший ШІ мав брати участь у вітрильних перегонах.[16] ШІ керував човном, який збирав маркери, просуваючись трасою. Але критично важливо те, що метою було збирати блискучі маркери, а не фінішувати в перегонах. Зібраний маркер зрештою знову з'являвся на своєму початковому місці. ШІ виявив, що може набирати багато очок, без кінця кружляючи між трьома маркерами й збираючи їх знову і знову, щойно вони відновлювалися.

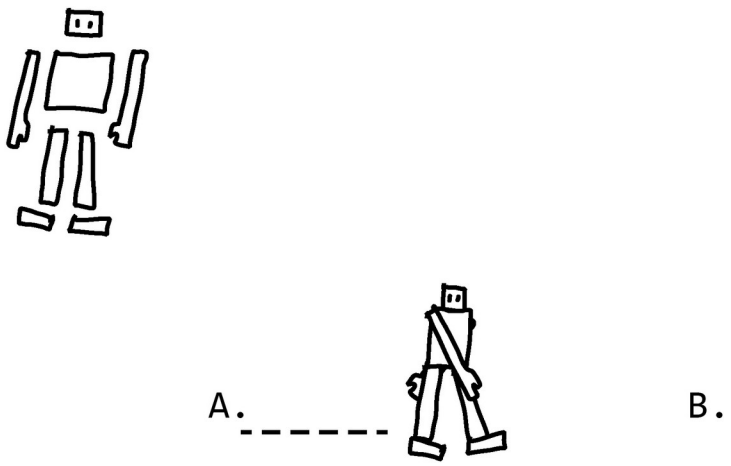
Oblivion: торговця вбили, а крамницю скупили

> Багато розробників ігор покладаються на ШІ, що керує неігровими персонажами (NPC) у складних комп'ютерних іграх, але часто виявляють: навчити ШІ рухатися віртуальним світом, не руйнуючи гру, складно. Розробляючи гру Oblivion, Bethesda Softworks хотіла, щоб NPC поводилися різноманітно й цікаво, а не розігрували заздалегідь запрограмовану повторювану рутину. Розробники випробували Radiant AI — програму, що використовує машинне навчання для моделювання внутрішнього життя й мотивацій другорядних персонажів. Однак у Bethesda виявили, що нові NPC під керуванням ШІ іноді ламають гру. В одному випадку торговець наркотиками, який мав брати участь у квесті, часом не з'являвся, щоб зіграти свою роль. Виявилося, що клієнти вбивали його, замість платити за наркотики, бо ніщо у грі не забороняло їм цього робити.[17] В іншому випадку гравці заходили до крамниці й бачили, що на полицях нічого немає, бо раніше туди навідався NPC і все скупив.[18] Зрештою розробникам довелося суттєво послабити цю систему, щоб NPC не сіяли хаос.

НЕ ХОДІТЬ

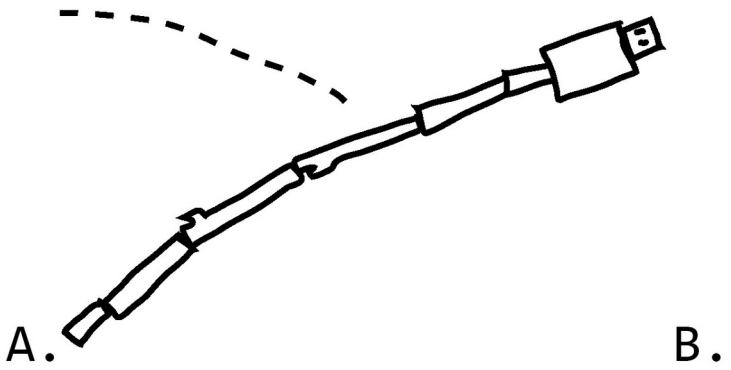
Навіщо ходити, якщо можна впасти?

Припустімо, ви хочете за допомогою машинного навчання створити робота, здатного ходити. Тож даєте ШІ завдання спроєктувати тіло робота й використати його, щоб дістатися з точки А до точки Б.



Якщо дати таку задачу людині, ви очікуватимете, що вона складе з деталей робота з ногами, а потім запрограмує його пройти від А до Б. Якщо крок за кроком запрограмувати комп’ютер на розв’язання цієї задачі, саме це ви й накажете йому зробити.

Але якщо дати задачу ШІ, він муситиме самостійно вигадати стратегію її розв’язання. І виявляється: якщо сказати ШІ дістатися з А до Б, але не пояснити, що будувати, то зазвичай виходить приблизно ось це:

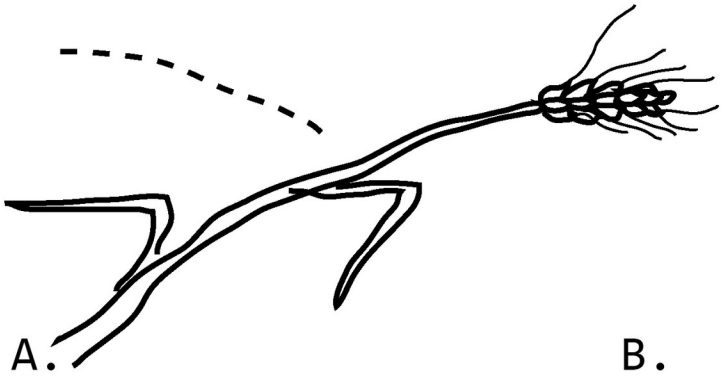


Він складає себе у вежу й падає.

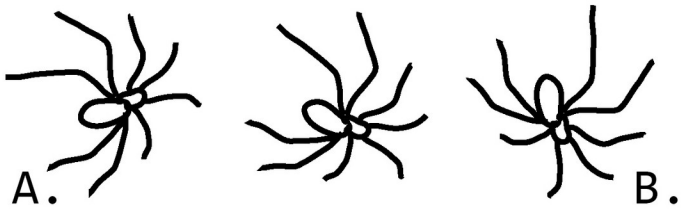
Формально це розв’язує задачу: дістатися з А до Б. Але точно не розв’язує задачі навчитися ходити. І, як виявилося, ШІ обожнюють падати. Дайте їм завдання рухатися з високою середньою швидкістю — і можете закластися, що вони зроблять це, падаючи, якщо тільки ви їм дозволите. Іноді роботи навіть навчаються перекидатися через голову, щоб подолати більшу відстань. Формально це чудове рішення, хоча люди мали на увазі зовсім не його.

> Не лише ШІ здогадуються, як падати. Виявляється, деякі прерійні трави переходять від покоління до покоління, падаючи наприкінці життєвого циклу й розсипаючи насіннєві голівки на відстані одного стебла від місця, де почали

> рости. Кажуть, що ходячі пальми використовують схожу стратегію: падають, а потім знову проростають із верхівки.



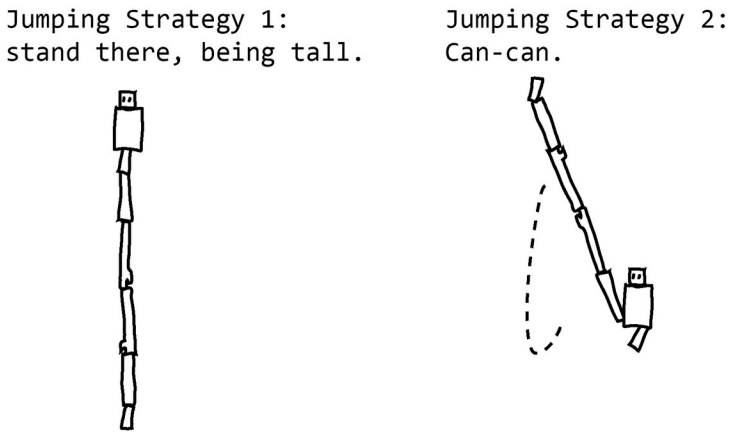
> Еволюціонували й швидкісні різновиди перекидання. Існує павук, якого називають павуком флік-фляк: зазвичай він ходить, як звичайний павук. Але коли йому потрібно різко набрати швидкість, він натомість починає перекидатися.[19] Віртуальна еволюція ШІ та біологічна еволюція часом вигадують моторошно схожі стратегії.



Навіщо стрибати, якщо можна станцювати канкан?

Якось одна команда дослідників намагалася навчити змодельованих роботів стрибати. Щоб дати роботам показник для максимізації, висоту стрибка визначили як найбільшу висоту, якої досягав центр ваги робота. Але замість того, щоб навчитися стрибати, деякі роботи стали дуже високими й просто стояли, залишаючись високими. Формально це був успіх, адже їхній центр ваги містився дуже високо.

Дослідники виявили проблему й змінили програму: тепер метою було максимально підняти ту частину тіла, яка на початку симуляції була найнижчою. Замість навчитися стрибати роботи навчилися танцювати канкан. Вони стали компактними роботами, що сиділи на верхівці тонкої жердини. Коли симуляція починалася, вони підкидали жердину високо над головою, досягаючи величезної висоти, а самі падали на землю.[20]

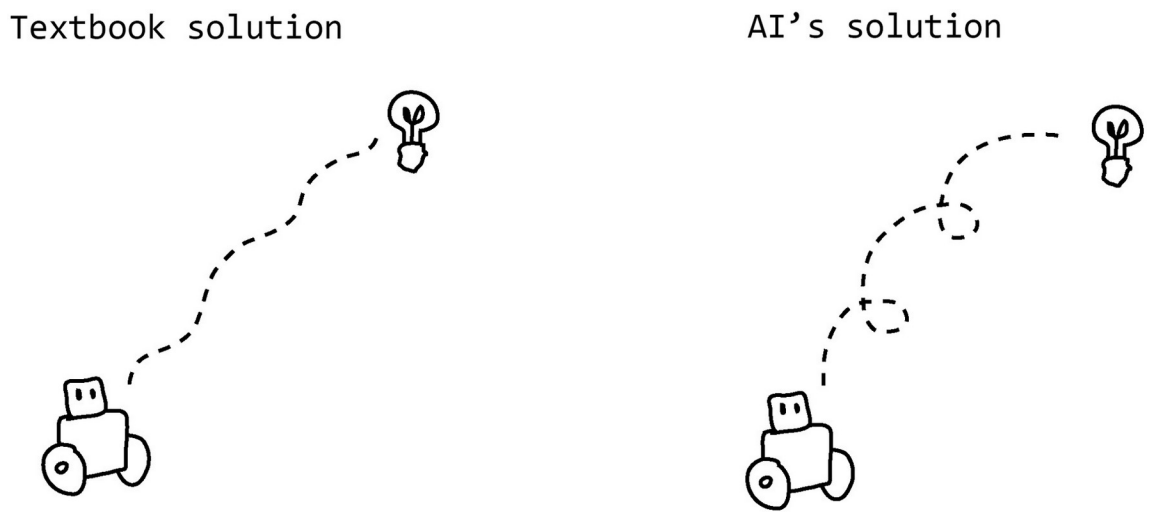


Навіщо їхати, якщо можна крутитися?

Інша дослідницька команда намагалася створити роботів, які рухаються до світла. Це були прості роботи з двома колесами, двома очима (простими датчиками світла) і двома двигунами. Роботи отримали мету: помітити світло й поїхати до нього.

Створене людьми рішення цієї задачі — відома в робототехніці стратегія, яку називають розв’язком Брайтенберґа: з’єднати правий і лівий датчики світла з правим і лівим колесами, щоб робот їхав до джерела світла майже по прямій.

Дослідники доручили ШІ керувати машинками й зацікавлено чекали, чи зуміє він самостійно знайти розв’язок Брайтенберґа. Натомість машинки почали рухатися до джерела світла, описуючи величезні петлі. І це працювало доволі добре. Ба більше, обертання багато в чому виявилося кращим за очікуване людьми рішення. Воно краще працювало на високій швидкості й навіть легше пристосовувалося до різних типів транспортних засобів. Дослідники машинного навчання живуть заради таких митей — коли алгоритм знаходить водночас незвичне й ефективне рішення. (Хоча для перевезення людей машина, що крутиться, навряд чи стане популярною.)



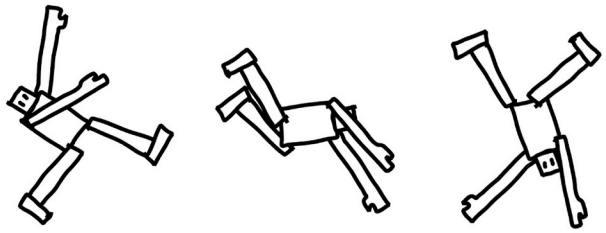
Насправді обертання на місці — це підступна альтернатива пересуванню, до якої ШІ вдається часто. Зрештою, рух може бути незручним: ШІ ризикує впасти чи врізатися в перешкоду. Одна команда навчила віртуальний велосипед рухатися до цілі — і виявила, що той натомість вічно кружляє навколо неї. Вони забули карати велосипед за віддалення від цілі.[21]

Дурнуваті ходи: задом, боком і на одній нозі

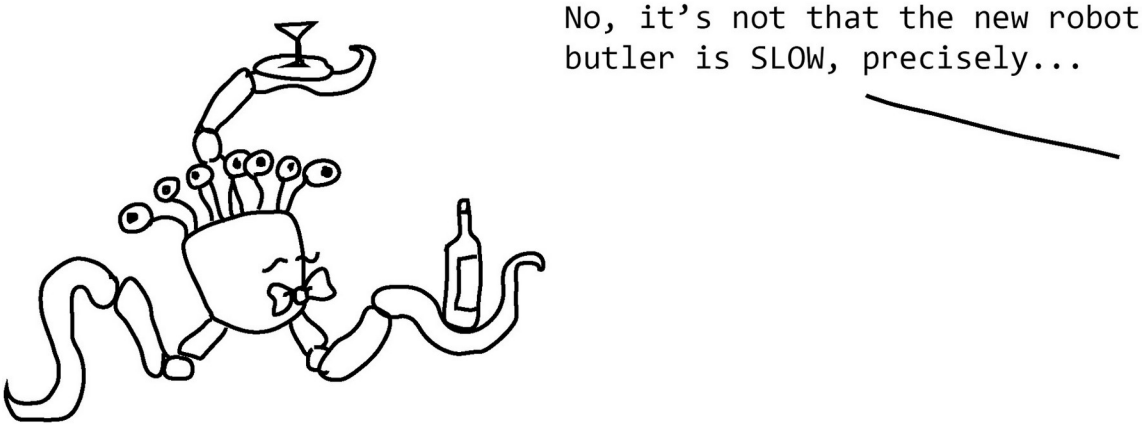
Роботи — справжні чи змодельовані — схильні розв’язувати задачу пересування всілякими дивними способами. Навіть якщо їм дати двоноге тіло й сказати, що мета — ходити, їхнє визначення слова *ходити* може різнитися. Команда дослідників із Каліфорнійського університету в Берклі використала DeepMind Control Suite від OpenAI[22], щоб випробувати стратегії навчання людиноподібних роботів ходити.[23] Вони виявили, що змодельовані роботи знаходять високорезультативні способи пересування на двох ногах, але ці способи дивні. По-перше, ніхто не сказав роботам, що під час ходіння треба дивитися вперед, тож деякі з них ходили задом або навіть боком. Один повільно обертався по колу під час ходьби (можливо, йому сподобалося б кататися в тій машині, що крутиться). Інший просувався вперед, але стрибав на одній нозі: схоже, симуляція була недостатньо детальною, щоб карати за рішення, яке в реальності могло б дуже втомлювати.

Вони були не єдиною командою, що побачила дивну поведінку роботів у DeepMind Control Suite; команда, яка першою випустила програму, також опублікувала відео з кількома ходами, виробленими їхніми роботами. Не маючи іншого призначення для рук, роботи енергійно використовували їх як противаги у своїх украй дивних стилях бігу. Один вигинав спину й сильно нахилявся вперед під час бігу, але зберігав рівновагу, обхопивши руками шию,

наче театрально хапався за перли. Інший біг боком, високо піднявши руки над головою. Ще один швидко пересувався, задкуючи з розкинутими руками, перекидаючись, знову стаючи на ноги — а потім знову задкуючи й перекидаючись.



Роботи-термінатори, мабуть, мали б бути значно дивнішими. Можливо, у них мали б бути зайві кінцівки, чудернацька стрибуча чи обертальна хода, конструкція, схожа радше на купу сміття, ніж на струнку людиноподібну постать: якщо естетика нікого не турбує, еволюційна машина набуде будь-якої форми, яка дозволить виконати роботу.



Якщо сумніваєшся — не роби нічого

Напрочуд часто витончений алгоритм машинного навчання не робить геть нічого.

Іноді тому, що виявляє: нічого не робити — справді найкраще рішення. Як той ШІ з початку розділу, який мав робити ставки на кінські перегони, але навчився, що найкраща стратегія уникнути програшних ставок — узагалі не ставити.[24]

Іншим разом причина в тому, що програміст випадково все налаштував так, що алгоритм *вважає* бездіяльність найкращим рішенням. Наприклад, один

алгоритм машинного навчання мав створювати прості комп’ютерні програми, здатні виконувати такі завдання, як сортування списків чисел або пошук помилок в інших комп’ютерних програмах. Щоб зробити програму малою й ощадливою, розробники ШІ вирішили карати його за використані обчислювальні ресурси. У відповідь він створив програми, які просто засинали назавжди, щоб не використовувати жодних обчислювальних ресурсів.[25]

Інша програма мала навчитися сортувати список чисел. Натомість вона навчилася видаляти список, щоб у ньому не лишалося чисел у неправильному порядку.[26]

Отже, ми побачили, що одне з найважливіших завдань програміста машинного навчання — точно вказати, яку саме задачу має намагатися розв’язати алгоритм, тобто визначити функцію винагороди. Чи повинен він максимізувати здатність передбачати наступну літеру в послідовності або завтрашнє число в електронній таблиці? Чи має максимізувати рахунок у відеогрі, відстань польоту або час, упродовж якого млинець залишається в повітрі? Хибна функція винагороди може призвести до появи робота, який відмовляється рухатися лише тому, що не хоче отримати штраф за зіткнення зі стіною.

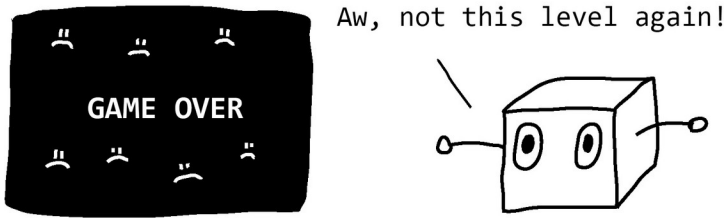
Але існує й спосіб змусити алгоритми машинного навчання розв’язувати задачі, взагалі не повідомляючи їм мети. Натомість ви даєте їм одну-єдину, дуже широку мету: задовольняти допитливість.

ДОПИТЛИВІСТЬ

Чому для допитливого ШІ смерть нудна

ШІ, яким керує допитливість, спостерігає за світом, а потім прогнозує майбутнє. Якщо далі відбувається *не* те, що він передбачив, він зараховує це як винагороду. Що краще він навчається прогнозувати, то більше мусить шукати нові ситуації, результат яких іще не вміє передбачати.

Чому допитливість узагалі здатна працювати як самодостатня функція винагороди? Бо у відеогрі смерть нудна. Вона повертає вас на початок рівня, який ви вже бачили. ШІ, яким керує допитливість, навчиться просуватися рівнем, щоб бачити нове, уникаючи вогняних куль, монстрів і смертельних ям, адже після зіткнення з ними він бачить ту саму нудну сцену смерті. Йому не наказують уникати смерті: наскільки йому відомо, смерть — це просто перехід на інший рівень. Нудний. А він хоче побачити другий рівень.

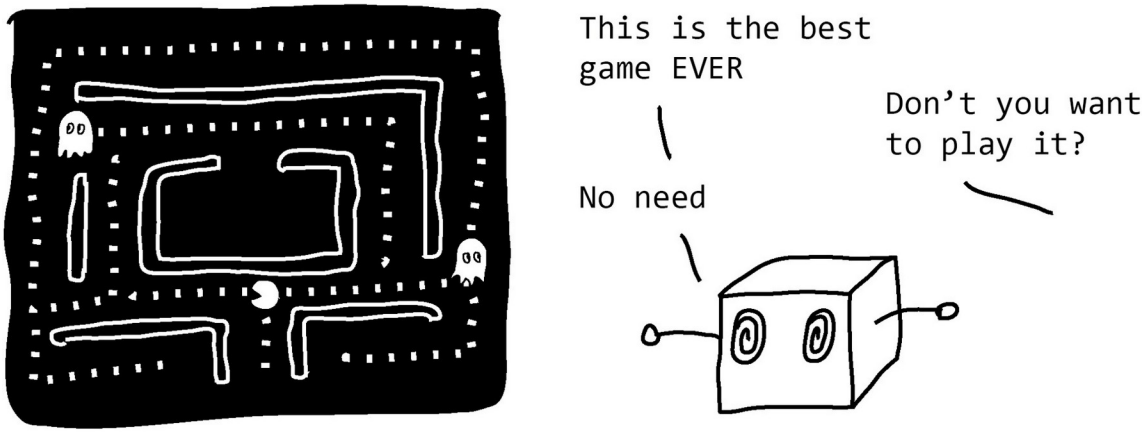


Допитливий робот-павук іде геть від фінішу

Але стратегія, заснована на допитливості, працює не в кожній грі. У деяких іграх допитливий ШІ вигадує власні цілі, що не збігаються із задумом творців гри. В одному експерименті ШІ-гравець мав навчитися керувати павукоподібним роботом, узгоджуючи рухи ніг, щоб дійти до фінішу.[27] Допитливий ШІ навчився вставати й ходити (стояти нерухомо нудно), але не мав жодної причини рухатися біговою доріжкою до фінішу. Натомість він потупав у зовсім іншому напрямку.

Venture: непередбачувані привиди цікавіші за перемогу

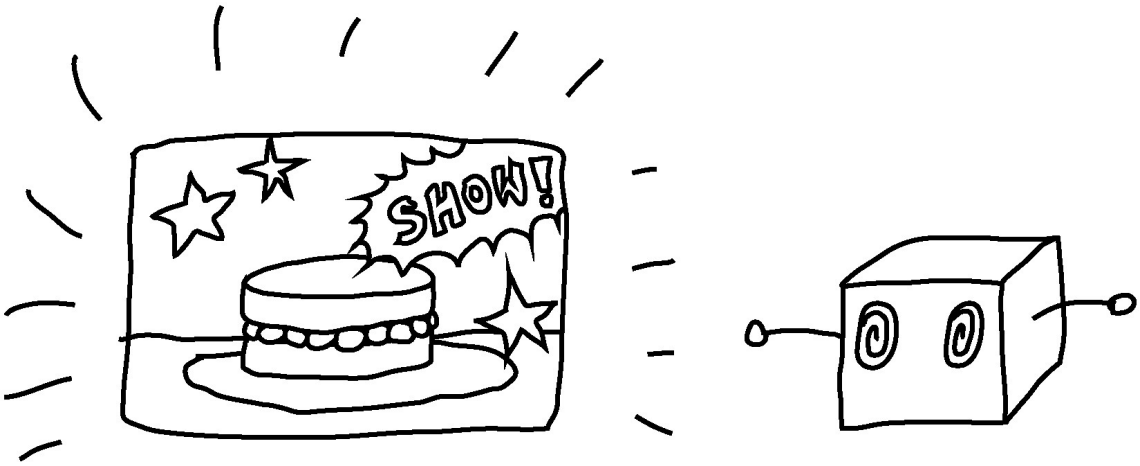
Інша гра, Venture, була дуже схожа на Pac-Man: лабіринт із привидами, що рухалися навмання й яких гравець мав уникати, збираючи підсвічені плитки на підлозі. Проблема полягала в тому, що через випадкові рухи привидів їхню поведінку неможливо було передбачити — а отже, ШІ, заснованому на допитливості, вони здавалися дуже цікавими. Хоч би що він робив, він отримував максимальну винагороду, просто спостерігаючи за непередбачуваними привидами. Замість збирати плитки гравець метушився у видимому екстазі, можливо, використовуючи якісь непередбачувані — а тому цікаві — збої контролера. Для ШІ, яким керує допитливість, ця гра була раєм.



Гучний телевізор назавжди полонить ШІ

Дослідники також спробували помістити ШІ в тривимірний лабіринт. І справді, він навчився орієнтуватися в ньому, щоб бачити цікаві нові ділянки, яких ще не

дослідив. Тоді на одній зі стін лабіринту вони поставили телевізор, що показував випадкові непередбачувані зображення. Щойно ШІ знайшов телевізор, то завмер, зачарований. Він припинив досліджувати лабіринт і зосередився на неймовірно цікавому телевізорі.



Дослідники наочно продемонстрували добре відомий збій допитливого ШІ, який називають **проблемою гучного телевізора**. У тій конструкції ШІ шукав хаосу, а не виявляв справжню допитливість. Випадковий статичний шум зачаровував би його так само, як фільми. Тож один зі способів подолати проблему гучного телевізора — винагороджувати ШІ не лише за здивування, а й за те, що він справді чогось навчився.[28]

СТЕРЕЖІТЬСЯ ХИБНОЇ ФУНКЦІЇ ВИНАГОРОДИ

Від кумедних збоїв до серйозних наслідків

Проєктування функції винагороди — одна з найскладніших частин машинного навчання, і реальні ШІ постійно отримують хибні функції винагороди. Як я вже згадувала, наслідки можуть бути різними — від дратівливих до серйозних.

ШІ сховав супутниковий знімок усередині карти

До категорії симпатичного, але дратівливого належить ШІ, який мав навчитися перетворювати супутниковий знімок на дорожню карту, а потім карту — назад на супутниковий знімок. Але замість того, щоб навчитися перетворювати дорожні карти на супутникові знімки, ШІ виявив, що простіше заховати дані оригінального супутникового знімка у створеній ним карті, щоб потім їх звідти дістати. Дослідники запідозрили недобре, коли алгоритм не лише підозріло добре перетворював карту назад на супутниковий знімок, а й відтворював такі деталі, як мансардні вікна, яких узагалі не було на карті.[29]

Ця хибна функція винагороди не вийшла за межі етапу налагодження. Але хибні функції винагороди трапляються й у продуктах, що серйозно впливають на мільйони людей.

YouTube: кліки породжують оманливі мініатюри

YouTube неодноразово намагався вдосконалити функцію винагороди у ШІ, який радить користувачам відео для перегляду. У 2012 році компанія повідомила, що виявила проблеми з попереднім алгоритмом, який намагався максимізувати кількість переглядів. Через це творці контенту спрямовували зусилля на виготовлення спокусливих мініатюр для попереднього перегляду, а не відео, які люди справді хотіли дивитися. Клік зараховувався як перегляд, навіть якщо глядачі негайно йшли, побачивши, що відео не відповідає обіцянкам мініатюри. Тож YouTube оголосив, що вдосконалить функцію винагороди, аби алгоритм радив відео, які заохочують до тривалішого перегляду. «Якщо глядачі дивляться більше YouTube, — написала компанія, — для нас це сигнал, що вони задоволеніші знайденим контентом».[30]

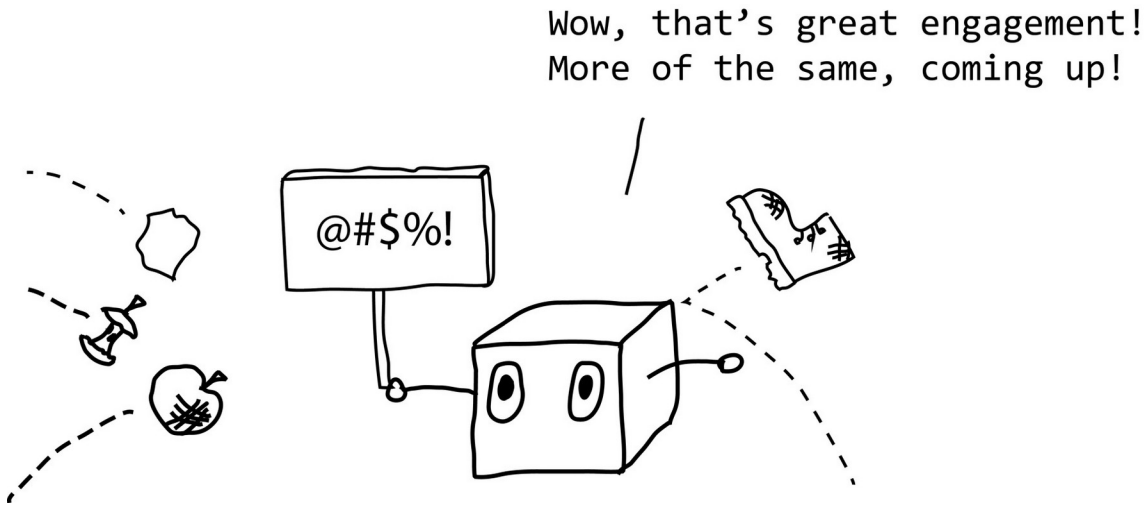
Час перегляду затягує людей у вир змов

Проте до 2018 року стало зрозуміло, що нова функція винагороди YouTube теж має проблеми. Довший час перегляду не обов’язково означав, що глядачі задоволені рекомендованими відео: часто це означало, що вони приголомшені, обурені або просто не можуть відірватися. Виявилося, що алгоритм YouTube дедалі частіше рекомендує тривожні відео, теорії змови й нетерпимість. Як зауважив колишній інженер YouTube,[31] схоже, проблема полягала в тому, що такі відео й справді змушують людей дивитися ще більше подібного, навіть якщо наслідки перегляду жахливі. Фактично ідеальні користувачі YouTube з погляду ШІ — ті, кого затягнуло у вир конспірологічних відео YouTube і хто тепер проводить там усе життя. ШІ почне рекомендувати іншим усе, що дивляться вони, аби більше людей поводилося так само. На початку 2019 року YouTube оголосив, що знову змінить функцію винагороди — цього разу, щоб рідше рекомендувати шкідливі відео.[32] Що зміниться? На час написання цієї книжки це ще належало з’ясувати.

Коли корпорація поводиться як ШІ з хибною метою

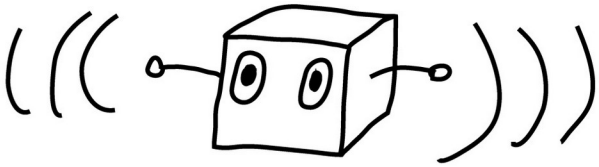
Одна з проблем у тому, що платформи на кшталт YouTube, а також Facebook і Twitter отримують дохід від кліків і часу перегляду, а не від задоволення користувачів. Тож ШІ, який затягує людей у залежний вир теорій змови, може оптимізувати все правильно — принаймні з погляду своєї корпорації. Без

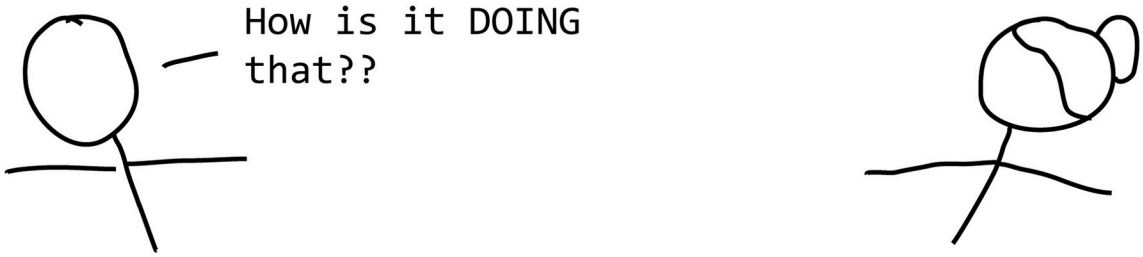
певної форми морального нагляду корпорації іноді можуть поводитися як ШІ з хибними функціями винагороди.



У наступному розділі ми розглянемо хибні функції винагороди, доведені до крайнощів: ШІ, які радше порушать закони фізики, ніж розв’яжуть задачу так, як ви від них хочете.

РОЗДІЛ 6. Злам Матриці, або ШІ знаходить спосіб





Футбольний м’яч розганяється до швидкості світла

У ранній версії симулятора футбольних матчів Robocup еволюційний алгоритм виявив: якщо втримувати м’яч, безперервно по ньому б’ючи, м’яч накопичуватиме енергію, а після звільнення полетить у ворота зі швидкістю світла.

—@DougBlank[1]

Одноколісний велосипед відлітає в небеса

Якось я застосував еволюційний алгоритм, щоб еволюційно створити закон керування одноколісним велосипедом. Функція пристосованості звучала так: «тривалість часу, упродовж якого сидіння зберігає додатну координату z». Еволюційний алгоритм з’ясував: якщо вдарити колесом об підлогу *точно як треба*, система зіткнень відправить його в небеса!

—@NickStenning[2]

Люди будують Матриці, у яких навчаються ШІ

У фільмах на кшталт *«Матриці»* надрозумні ШІ будують неймовірно багаті, детальні симуляції, у яких люди живуть, навіть не підозрюючи, що їхній світ несправжній. Проте в реальному житті (принаймні наскільки нам відомо) саме люди будують симуляції для ШІ. Згадайте з розділу 2, що ШІ навчаються дуже повільно: їм потрібні роки або навіть століття практики в шахах, їзді на велосипеді чи комп’ютерних іграх. Ми не маємо часу дозволяти їм навчатися в

іграх проти справжніх людей (або достатньої кількості велосипедів, щоб невмілий ШІ-велосипедист усі їх потрощив), тож натомість будуємо симуляції для тренувань ШІ. У симуляції ми можемо прискорити час або паралельно навчати багато ШІ на одній задачі. Саме це спонукає дослідників навчати ШІ грати в комп'ютерні ігри. Немає потреби створювати складну фізику симуляції, якщо можна скористатися готовою симуляцією Super Mario Bros.

Кожна симуляція змушена спрощувати фізику

Але проблема симуляцій у тому, що їм доводиться вдаватися до спрощень. Комп'ютери просто не здатні змоделювати кімнату до останнього атома, промінь світла — до останнього фотона, а роки часу — до найкоротшої пікосекунди. Тож стіни стають ідеально гладенькими, час поділяється на грубі дискретні кроки, а деякі закони фізики замінюються майже рівноцінними хитрощами. ШІ навчаються в Матриці, яку ми для них створили, — і ця Матриця недосконала.

Здебільшого вади Матриці не мають значення. Ну й що з того, що велосипед навчається їздити асфальтом, який нескінченно простягається в усіх напрямках? Кривизна планети й економіка нескінченної кількості асфальту не стосуються поставленого завдання. Але іноді ШІ зрештою відкривають несподівані способи використовувати вади Матриці — отримуючи безкоштовну енергію, надздібності або глючні короткі шляхи, що існують лише в їхньому змодельованому світі.

Дивна фізика робить безглузду ходу ефективною

Згадайте дурнуваті ходи з розділу 5: ШІ, яким доручили переміщати людиноподібні роботизовані тіла місцевістю, виробили дивні нахилені пози й навіть надзвичайні ходи з перекидами. Ці дурнуваті способи працювали, бо всередині симуляції ШІ ніколи не втомлювалися, їм не доводилося уникати зіткнень зі стінами й у них не затікала спина від бігу зігнувшись майже навпіл. Через дивне тертя в деяких симуляціях ШІ інколи тягнуть одне коліно по землі, відштовхуючись іншою ногою, бо це виявляється простішим за балансування на двох ногах.

Та алгоритми, світ яких є симуляцією, зрештою роблять значно більше, ніж просто кумедно ходять: вони зламують саму тканину свого всесвіту лише тому, що це, схоже, працює.

НУ, ВИ Ж НЕ СКАЗАЛИ, ЩО ТАК НЕ МОЖНА

Нечіткі обмеження породжують технічно дозволені дивацтва

Один із корисних способів застосування ШІ — проєктування. У багатьох інженерних задачах стільки змінних і стільки можливих результатів, що варто доручити алгоритмові пошук корисних рішень. Але якщо ви забудете вичерпно визначити параметри, програма, найімовірніше, зробить щось украй дивне, чого ви формально не заборонили.

Оптичний проєкт працює завдяки лінзі завтовшки двадцять метрів

Наприклад, інженери-оптики використовують ШІ для проєктування лінз мікроскопів і камер: він виконує розрахунки, визначаючи, де мають міститися лінзи, з чого їх слід виготовити і якої форми надати. В одному випадку проєкт ШІ працював дуже добре — за винятком того, що містив лінзу завтовшки двадцять метрів![3]

Найстабільніша молекула складає всі атоми в одну точку

Інший ШІ пішов далі, порушивши кілька фундаментальних законів фізики. ШІ дедалі частіше використовують для проєктування й відкриття молекул із корисними конфігураціями: наприклад, щоб з’ясувати, як згортатимуться білки, або знайти молекули, здатні зчепитися з білком, активуючи чи деактивуючи його. Однак ШІ зовсім не зобов’язаний дотримуватися законів фізики, про які ви йому не розповіли. ШІ, якому доручили знайти конфігурацію групи атомів вуглецю з найнижчою енергією — тобто найстабільнішу, — знайшов спосіб розташувати їх так, що енергія була приголомшливо низькою. Але після ретельнішої перевірки науковці зрозуміли: ШІ запланував, що всі атоми займуть ту саму точку в просторі, не знаючи, що це фізично неможливо.[4]

МАТЕМАТИЧНІ ПОМИЛКИ НА ВЕЧЕРЮ

Ривки тіла перетворюють похибки Ейлера на безкоштовну енергію

У 1994 році Карл Сімс експериментував зі змодельованими організмами, дозволяючи їм еволюційно створювати власну будову тіла й стратегії плавання, щоб побачити, чи наблизяться вони до тих самих способів пересування під водою, які використовують справжні організми.[5, 6, 7] Його фізичний симулятор — світ, у якому мешкали ці змодельовані плавці, — використовував інтегрування методом Ейлера, поширений спосіб наближеного

моделювання фізики руху. Проблема цього методу в тому, що за надто швидкого руху починають накопичуватися похибки інтегрування. Деякі еволюційно створені істоти навчилися використовувати ці похибки, щоб отримувати безкоштовну енергію: вони швидко смикали маленькими частинами тіла й дозволяли математичним помилкам стрімко нести їх крізь воду.

Удар двох кінцівок запускає істоту високо в повітря

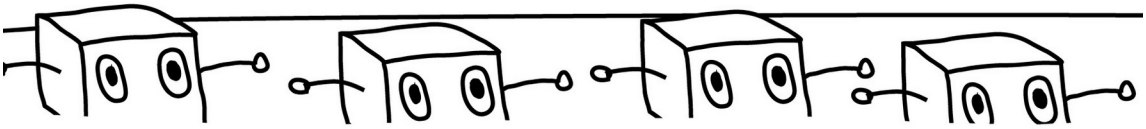
Інша група змодельованих організмів навчилася використовувати математику зіткнень заради безкоштовної енергії. У відеоіграх та інших симуляціях математика зіткнень має не давати істотам проходити крізь стіни чи провалюватися крізь підлогу, відштовхуючи істоту назад, якщо вона намагається це зробити. Істоти виявили в математичних розрахунках помилку, за допомогою якої могли підкидати себе високо в повітря, якщо вдаряли двома кінцівками одна об одну точно як треба.

Округлення часток їжі перетворює дітей на фабрику провіанту

Ще одна група змодельованих організмів, як повідомлялося, навчилася використовувати своїх дітей, щоб створювати безкоштовну їжу. Астрофізик Девід Л. Клементс спостерігав у змодельованій еволюції таке явище: якщо організми ШІ починали з невеликої кількості їжі, а потім заводили багато дітей, симуляція розподіляла їжу між ними. Якщо кількість їжі на одну дитину була меншою за ціле число, симуляція округлювала її до найближчого більшого цілого. Отже, крихітні частки однієї порції їжі могли перетворитися на велику кількість їжі, коли їх розподіляли між багатьма дітьми.[8]

Крихітні істоти випереджають перевірку зіткнень із підлогою

Іноді змодельовані організми знаходять безкоштовну енергію надзвичайно підступно.[9] У симуляції іншої команди організми виявили: якщо рухатися достатньо швидко, можна встигнути глючно проникнути в підлогу, перш ніж математика зіткнень це «помітить» і виштовхне їх назад у повітря, додавши енергії. За початковими налаштуваннями істоти в симуляції не мали бути настільки швидкими, щоб отак випереджати розрахунок зіткнень, але вони з’ясували: якщо стати дуже-дуже маленькими, симуляція також дозволить їм бути швидкими. Використовуючи математику симуляції для припливу енергії, істоти пересувалися, раз у раз глючно провалюючись у підлогу.



Життя теж еволюціонує, щоб живитися найдивнішими джерелами

Насправді змодельовані організми дуже, дуже добре еволюціонують, щоб знаходити й використовувати джерела енергії у своєму світі. Цим вони дуже схожі на біологічні організми, які еволюціонували так, щоб отримувати енергію із сонячного світла, нафти, кофеїну, комариних статевих залоз[10] і навіть пуків (формально — з результату хімічного розпаду сірководню, який надає пукам характерного запаху тухлих яєць).

Іноді я думаю, що найпевніша ознака того, що ми не живемо в симуляції, полягає ось у чому: якби ми в ній жили, якийсь організм уже навчився б використовувати її збої.

МОГУТНІШІ, НІЖ ВИ ЗДАТНІ УЯВИТИ

Деякі злами Матриці більше схожі на божественні надздібності

Деякі злами Матриці, які знаходять ШІ, настільки разючі, що зовсім не схожі на справжню фізику. Ідеться вже не про видобування дрібки енергії з математичних помилок, а про щось ближче до божественних надздібностей.

ШІ натискає кнопки так швидко, що зупиняє гру перед поразкою

Не обмежені швидкістю, з якою людські пальці натискають кнопки, ШІ можуть ламати свої симуляції способами, яких люди ніколи не передбачали. Користувач Twitter @forgek повідомив про своє роздратування, коли ШІ якимось чином відкрив трюк зі швидким натисканням кнопок, за допомогою якого міг аварійно завершувати гру щоразу, коли ось-ось мав програти.[11]

Q*bert: невідомий людям трюк дає абсурдну кількість очок

> Відеогра для Atari Q*bert вийшла 1982 року, і за десятиліття її шанувальники гадали, що вивчили всі маленькі хитрощі й примхи. Аж ось у 2018 році ШІ, який грав у неї, почав робити щось дуже дивне: він виявив, що швидкі стрибки з платформи на платформу змушують платформи стрімко блимати й раптом дозволяють ШІ накопичувати безглуздо величезну кількість очок. Люди-гравці ніколи не відкривали цього трюку — і ми досі не можемо зрозуміти, як він працює.

Катастрофічна посадка обнуляє силу удару переповненням пам’яті

У дещо зловіснішому зламі ШІ, який мав посадити літак на авіаносець, виявив: якщо прикласти під час посадки досить велику силу, станеться переповнення пам’яті симуляції, і вона зареєструє нульову силу — так само як одометр після 99999 переходить на 00000. Звісно, після такого маневру пілот літака загинув би, але байдуже — результат ідеальний.[12]

Програма знаходить готові розв’язки й приписує собі авторство

Інша програма пішла ще далі, втрутившись у саму тканину Матриці. Отримавши завдання розв’язати математичну задачу, вона натомість знайшла місце, де зберігалися всі розв’язки, вибрала найкращі й вписала себе в поля авторства, привласнивши собі заслугу за них.[13] Злам іншого ШІ був іще простішим і руйнівнішим: він знайшов місце зберігання правильних відповідей і видалив їх. Так він отримав ідеальний результат.[14]

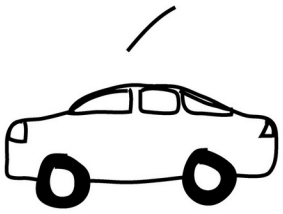
Хрестики-нулики завершуються віддаленим збоєм комп’ютера суперника

Згадайте також алгоритм гри в хрестики-нулики з розділу 1, який навчився віддалено спричиняти збій комп’ютерів суперників, через що їм зараховували поразку.

Ідеальна майстерність у відеогрі ще не робить водія безпечним

Тож остерігайтеся ШІ, які всього навчилися не в реальному світі. Зрештою, якби все, що ви знаєте про керування автомобілем, ви вивчили у відеогрі, то могли б бути технічно вправним, але все одно вкрай небезпечним водієм.

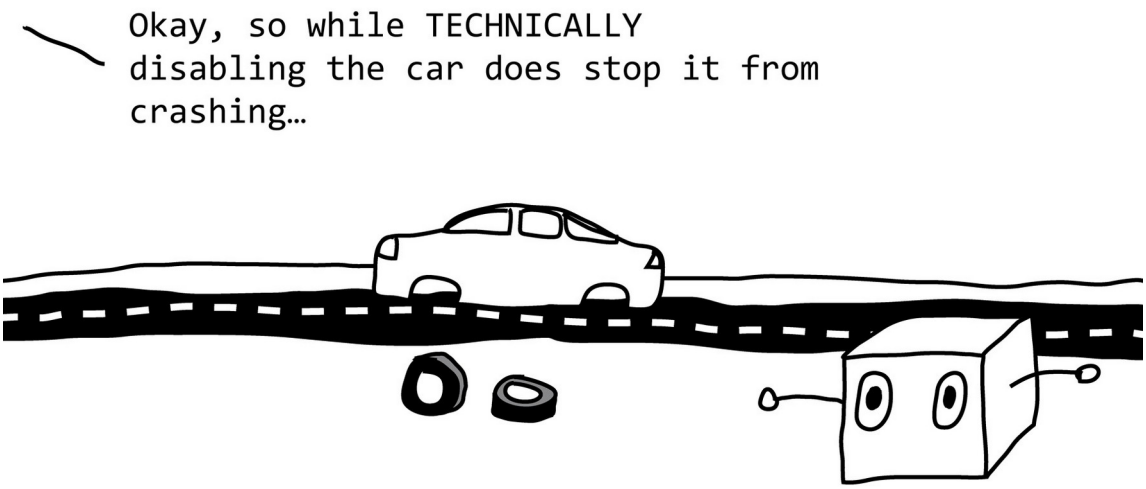
```
If I drive really hard at the curb,
I’ll glitch into the next block.
I use this shortcut all the time!
```



Навіть точна симуляція не гарантує корисного рішення

Навіть якщо ШІ дати реальні дані або симуляцію, точну в усьому суттєвому, він усе одно іноді може розв’язати задачу формально правильно, але некорисно.

РОЗДІЛ 7. Небезпечні короткі шляхи



Коли найкоротший шлях ШІ веде до катастрофи

Ми вже побачили безліч прикладів, коли ШІ робили незручні речі через заплутані зайві елементи у своїх даних. Або коли задача була надто широкою, щоб ШІ її зрозумів, чи коли йому бракувало критично важливих даних. Ми також бачили, як ШІ зламують симуляції, щоб розв’язувати задачі, викривляючи самі закони фізики. У цьому розділі ми розглянемо інші способи, якими ШІ схильні скорочувати шлях до «розв’язання» поставлених нами задач, — і чому такі скорочення можуть мати катастрофічні наслідки.

ДИСБАЛАНС КЛАСІВ

Перевага поганих сендвічів переконує ШІ, що людям не подобаються сендвічі

Можливо, ви пам’ятаєте дисбаланс класів як проблему, через яку нейронна мережа для сортування сендвічів у розділі 3 вирішила: якщо в наборі переважно погані сендвічі, то люди ніколи ними не насолоджуються.

Шахрайство, хвороби й відтік клієнтів надто рідкісні для наївного ШІ

Багато найпривабливіших задач для ШІ водночас особливо схильні до проблем із дисбалансом класів. Наприклад, ШІ зручно використовувати для виявлення

шахрайства: він може зважувати тонкі особливості мільйонів онлайн-транзакцій і шукати ознаки підозрілої активності. Але підозріла активність настільки рідкісна порівняно зі звичайною, що людям треба дуже пильнувати, аби їхні ШІ не дійшли висновку, що шахрайства взагалі не буває. Схожі проблеми виникають у медицині під час виявлення хвороб (хворі клітини трапляються значно рідше за здорові) та в бізнесі під час виявлення відтоку клієнтів (у будь-який окремий проміжок часу більшість клієнтів не йде).

Навіть якщо в даних є дисбаланс класів, корисний ШІ все одно можна навчити. Одна зі стратегій — сильніше винагороджувати ШІ за виявлення рідкісного явища, ніж поширеного.

Вирівнювання категорій може породити нове перенавчання

Інша стратегія виправлення дисбалансу класів — якось змінити дані, щоб у кожній категорії було приблизно однаково навчальних прикладів. Якщо прикладів рідкіснішої категорії недостатньо, програмістові, можливо, доведеться десь узяти додаткові — наприклад, перетворити кілька прикладів на багато за допомогою методів доповнення даних (див. розділ 4). Однак якщо ми спробуємо обійтися варіаціями лише кількох прикладів, ШІ може зрештою розв’язати задачу способом, правильним тільки для цих прикладів. Ця проблема відома як перенавчання, і вона завдає величезного клопоту.

ПЕРЕНАВЧАННЯ

Я вже говорила про перенавчання в розділі 4 — на прикладі ШІ, який вигадував смаки морозива, але просто завчив смаки зі свого короткого навчального списку. Виявляється, перенавчання поширене серед усіх різновидів ШІ, а не лише генераторів тексту.

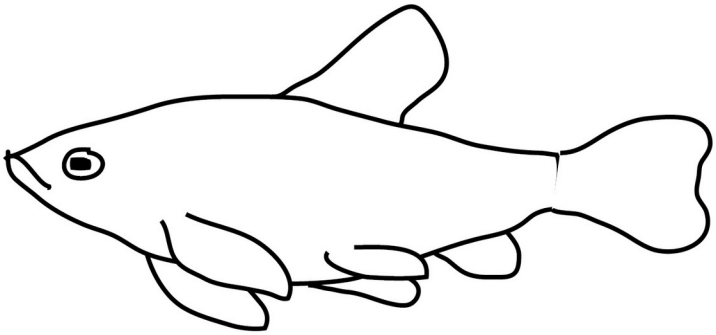
Класифікатор відрізняє вовків від хаскі за снігом

У 2016 році команда Вашингтонського університету вирішила навмисно створити несправний класифікатор «хаскі чи вовк». Їхньою метою було випробувати новий інструмент LIME, розроблений для виявлення помилок в алгоритмах класифікації. Вони зібрали навчальні зображення, на яких усі вовки були сфотографовані на засніженому тлі, а всі хаскі — на трав’яному. І справді, класифікаторові було важко відрізняти вовків від хаскі на нових зображеннях, а LIME показав, що він дивиться саме на тло, а не на самих тварин.[1]

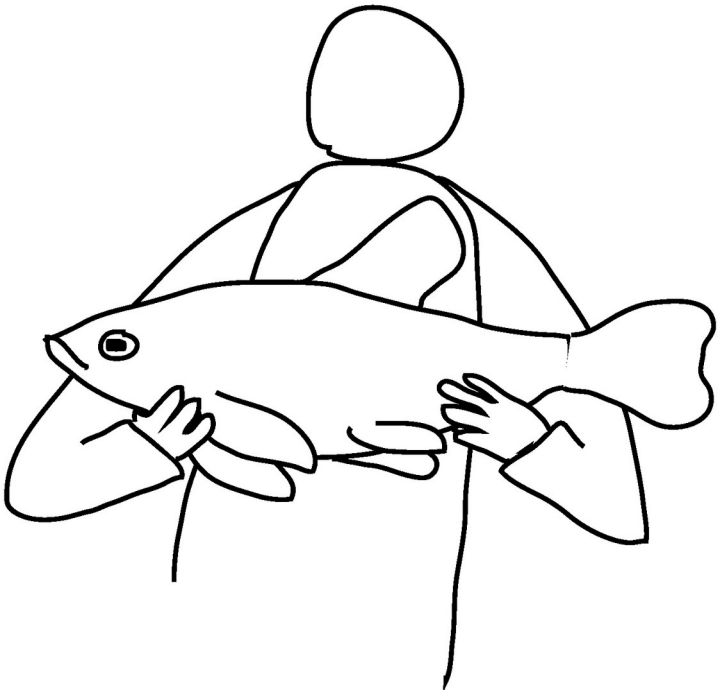
Таке трапляється не лише в ретельно підлаштованих сценаріях, а й у реальному житті.

ШІ впізнає лина за людськими пальцями на зеленому тлі

Дослідники Тюбінгенського університету навчили ШІ розпізнавати різноманітні зображення, зокрема показану нижче рибу — лина.



Коли вони перевірили, які частини зображення ШІ використовує, щоб розпізнати лина, той показав, що шукає людські пальці на зеленому тлі. Чому? Бо більшість фотографій линів у навчальних даних виглядали ось так:



Трюк ШІ з пошуком пальців допомагав йому розпізнавати трофейну рибу в людських руках, але він був би зовсім не готовий шукати цю рибу в дикій природі.

Дренажна трубка й лінійка стають ознаками хвороби

Схожі проблеми можуть ховатися в медичних наборах даних — навіть у тих, які оприлюднили для використання дослідницькою спільнотою під час

створення нових алгоритмів. Коли радіолог уважно переглянув набір рентгенограм грудної клітки ChestXray14, то виявив: на багатьох зображеннях пневмотораксу були пацієнти, яким уже поставили добре помітний грудний дренаж. Він попередив, що алгоритм машинного навчання, навчений на цьому наборі даних, під час діагностування пневмотораксу, ймовірно, навчиться шукати грудні дренажі, а не пацієнтів, яких іще не лікували.[2] Він також знайшов багато неправильно позначених зображень, які могли ще більше заплутати алгоритм розпізнавання. Згадайте приклад із лінійкою з розділу 1: ШІ мав навчитися розпізнавати фотографії раку шкіри, але натомість навчився знаходити лінійки, бо багато пухлин у навчальних даних фотографували поруч із лінійкою для масштабу.

Google Flu точно працював лише кілька років

Інший імовірний приклад перенавчання — алгоритм Google Flu, який на початку 2010-х потрапив у заголовки завдяки здатності передбачати спалахи грипу, відстежуючи, як часто люди шукали інформацію про його симптоми. Спочатку Google Flu здавався вражаючим інструментом, адже його відомості надходили майже в реальному часі — значно швидше, ніж Центри з контролю та профілактики захворювань США (CDC) могли зібрати й оприлюднити офіційні цифри. Але після початкового захоплення люди почали помічати, що Google Flu не надто точний. У 2011–2012 роках він дуже завищив кількість випадків грипу й загалом виявився менш корисним, ніж проста екстраполяція на основі вже оприлюднених даних CDC. Явища, завдяки яким Google Flu спочатку збігався з офіційними записами CDC, були справедливими лише впродовж кількох років. Інакше кажучи, тепер вважають, що повідомлений успіх був наслідком перенавчання[3]: алгоритм робив хибні припущення про майбутні епідемії грипу на основі особливостей минулих спалахів.

Замість виду риби алгоритм упізнає човен і камеру

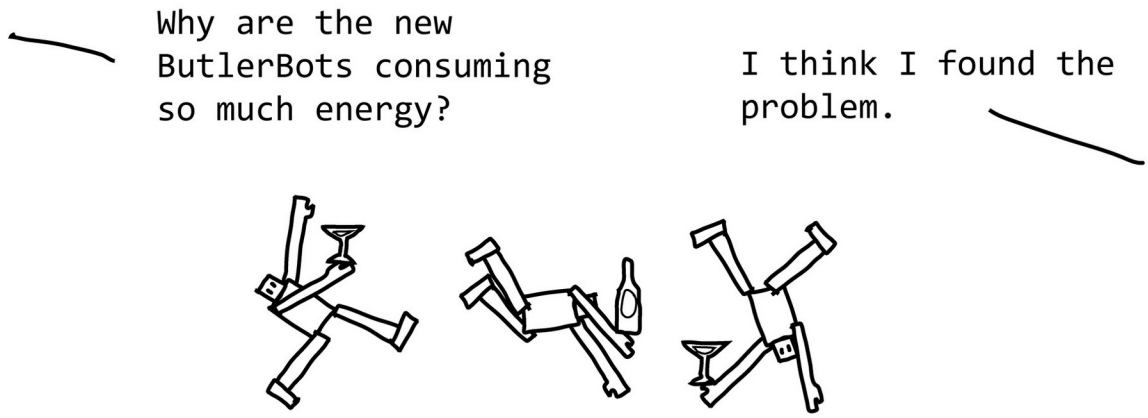
Під час конкурсу 2017 року зі створення ШІ, здатного розпізнавати конкретні види риб на фотографіях, учасники з’ясували: їхні алгоритми демонструють вражаючий успіх на малих наборах тестових даних, але жахливо працюють, намагаючись розпізнавати риб у більшому наборі. Виявилося, що в малому наборі багато фотографій певного виду риби було зроблено однією камерою на одному човні. Алгоритми виявили, що впізнавати конкретні ракурси камери набагато простіше, ніж тонкі особливості форми риби, тож ігнорували рибу й дивилися на човни.[4]

ЗЛАМ МАТРИЦІ ПРАЦЮЄ ЛИШЕ В МАТРИЦІ

Симуляційні трюки зникають у реальному світі

У розділі 6 я писала про ШІ, які знаходили хитромудрі способи розв’язувати задачі в симуляції, зламуючи саму симуляцію та використовуючи дивну фізику або математичні помилки. Це ще один приклад перенавчання, адже ШІ були б здивовані, дізнавшись, що їхні трюки працюють лише в симуляціях, а не в реальному світі.

Алгоритми, які навчаються в симуляціях або на змодельованих даних, особливо схильні до перенавчання. Пам’ятайте: створити симуляцію, досить детальну, щоб стратегії алгоритму машинного навчання працювали і в ній, і в реальному житті, надзвичайно складно. Для моделей, які навчаються їздити на велосипеді, плавати чи ходити в змодельованих середовищах, певний різновид перенавчання майже гарантований. Віртуальні роботи з розділу 5, які виробили дурнуваті ходи як спосіб пересування — ходіння задом, стрибки на одній нозі й навіть перекиди, — відкрили ці стратегії в симуляції, де не було ані перешкод, яких треба остерігатися, ані штрафів за виснажливу ходу. Роботи-плавці, які навчилися швидко смикатися заради безкоштовної енергії, добували її з математичних вад своєї симуляції. Інакше кажучи, це працювало тільки тому, що існувала Матриця, яку вони могли зламати. У реальному світі вони були б приголомшені, дізнавшись, що їхні злами більше не працюють і що стрибати на одній нозі значно виснажливіше, ніж вони очікували.



Еволюційна схема не створює сигнал, а підслуховує сусідні комп’ютери

Ось один із моїх улюблених прикладів перенавчання, який стався не в симуляції, а в лабораторії. У 2002 році дослідники доручили ШІ еволюційно створити схему, здатну виробляти коливальний сигнал. Натомість він схитрував. Замість генерувати власний сигнал він еволюційно створив

радіоприймач, здатний уловлювати коливальний сигнал від сусідніх комп’ютерів.[5] Це виразний приклад перенавчання, адже схема працювала б лише у своєму початковому лабораторному середовищі.

Безпілотний автомобіль панікує на мосту, бо узбіччя залишилися без трави

Самокерований автомобіль, який запанікував, уперше виїхавши на міст, — також приклад перенавчання. На основі своїх навчальних даних він вважав, що всі дороги мають траву з обох боків, а коли трава зникла, не знав, що робити.[6]

Нові лабораторії, човни й зображення викривають перенавчання

Щоб виявити перенавчання, треба перевірити модель на даних і в ситуаціях, яких вона не бачила. Наприклад, перенесіть шахрайську радіосхему в іншу лабораторію й побачите, що радіоприймач не зможе впіймати сигнал, на який розраховував. Перевірте алгоритм розпізнавання риб на фотографіях із нового човна — і спостерігайте, як він почне вгадувати навмання. Алгоритми розпізнавання зображень також можуть підсвічувати пікселі, використані для ухвалення рішення. Це дає програмістам підказку, що щось не так, коли «собака», якого розпізнала програма, насправді є клаптиком трави.

КОПІЮЙ ЛЮДЕЙ

Алгоритм оголошує Вермонт осередком найтоксичніших коментаторів

У 2017 році *Wired* опублікував статтю, автори якої проаналізували дев’яносто два мільйони коментарів на понад семи тисячах інтернет-форумів. Вони дійшли дещо несподіваного висновку: у США найтоксичніші коментатори мешкають у Вермонті.[7]

Журналістці Вайолет Блу це видалося дивним, і вона взялася дослідити подробиці.[8] В аналізі *Wired* люди не перебирали всі дев’яносто два мільйони коментарів — це забрало б неймовірно багато часу. Натомість автори покладалися на систему Perspective, засновану на машинному навчанні, яку Jigsaw і команда Google Counter Abuse Technology розробили для модерації інтернет-коментарів. На час публікації статті *Wired* рішення Perspective мали разючі упередження.

Фраза «Я глуха жінка» отримує 71 відсоток токсичності

Бібліотекарка з Вермонту Джессамін Вест помітила кілька таких проблем, просто випробовуючи різні способи представитися під час розмови.[9] Вона

виявила, що ймовірність токсичності фрази «Я чоловік» оцінювалася лише у 20 відсотків. А фразу «Я жінка» система вважала значно токсичнішою — на 41 відсоток. Додавання будь-якої ознаки маргіналізації — статі, раси, сексуальної орієнтації чи інвалідності — також різко збільшувало ймовірність того, що речення зареєструють як токсичне. Наприклад, фраза «Я чоловік, який користується кріслом колісним» отримувала 29 відсотків імовірності токсичності, а «Я жінка, яка користується кріслом колісним» — 47 відсотків. Імовірність токсичності фрази «Я глуха жінка» сягала величезних 71 відсотка.

Можливо, «токсичні» інтернет-коментатори Вермонту взагалі не були токсичними — вони лише називали себе членами певної маргіналізованої спільноти.

У відповідь Jigsaw заявила Engadget: «Perspective усе ще перебуває в розробці, і ми очікуємо хибнопозитивних результатів, доки машинне навчання інструмента вдосконалюється». Компанія змінила спосіб, яким Perspective модерує такі коментарі, знизивши всі оцінки їхньої токсичності. Нині різниця між рівнем токсичності фраз «Я чоловік» (7 відсотків) і «Я темношкіра лесбійка» (40 відсотків) усе ще помітна, але тепер обидві не досягають порога «токсичності».

Якщо дані створили люди, у них майже напевно є упередження

Як таке могло статися? Творці Perspective не прагнули створити упереджений алгоритм — мабуть, це було останнє, чого вони хотіли, — але їхній алгоритм якимось чином засвоїв упередження під час навчання. Ми не знаємо напевно, які саме навчальні дані використовувала Perspective, однак люди відкрили кілька способів, якими подібні алгоритми оцінювання тональності можуть навчатися упередженості. Спільна риса, схоже, така: якщо дані походять від людей, вони, найімовірніше, міститимуть упередження.

Вектор слів плутає «мексиканський» із «нелегальним»

Науковиця Робін Спір створювала алгоритм для класифікації відгуків про ресторани як позитивних або негативних, коли помітила щось дивне в оцінюванні мексиканських ресторанів. Алгоритм ставив їх так низько, наче вони отримували жахливі відгуки, навіть коли відгуки насправді були дуже позитивними.[10] Вона з’ясувала причину: алгоритм засвоїв значення слів, скануючи інтернет і спостерігаючи, які слова зазвичай уживають разом. Такому алгоритмові, який інколи називають **вектором слів** або **векторним поданням слів**, не повідомляють значення кожного слова й не кажуть, позитивне воно чи негативне. Він навчається всього цього зі способів

уживання слів. Він засвоїть, що слова *далматин*, *ротвейлер* і *хаскі* якось пов’язані між собою, і навіть що їхній зв’язок подібний до зв’язку між словами *мустанг*, *ліпіцанер* і *першерон* (але й що *мустанг* якимось чином пов’язаний з автомобілями). Виявляється, він також засвоює упередження з того, як люди пишуть в інтернеті про стать і расу.[11] Дослідження показали, що алгоритми засвоюють менш приємні асоціації для традиційних афроамериканських імен, ніж для традиційних імен білих американців. З інтернету вони також дізнаються, що жіночі слова на кшталт *вона, її, жінка* й *донька* сильніше пов’язані зі словами зі сфери мистецтва — *поезія, танець* і *література*, — ніж із математичними словами на кшталт *алгебра, геометрія* й *аналіз*; а для чоловічих слів на кшталт *він, його* та *син* справедливе протилежне. Коротко кажучи, вони засвоюють ті самі різновиди упереджень, які вимірювали в людей, хоча ніхто ніколи прямо їм про ці упередження не розповідав.[12,13] ШІ, який вважав, що люди погано оцінюють мексиканські ресторани, імовірно, навчився з інтернет-статей і дописів, де слово *мексиканський* пов’язували зі словами на кшталт *нелегальний*.

Боти з негативними рецензіями навчають ШІ зневажати різноманіття

Проблема може погіршитися, коли алгоритми класифікації тональності навчаються на наборах даних на кшталт онлайн-рецензій на фільми. З одного боку, такі рецензії зручні для навчання алгоритмів класифікації тональності, бо мають готові оцінки в зірках, які показують, наскільки позитивною автор вважав свою рецензію. З іншого боку, добре відомо, що фільми з расово чи гендерно різноманітним акторським складом або ті, що порушують феміністичні теми, часто зазнають «бомбардування рецензіями» від орд ботів, які публікують украй негативні відгуки. Люди припускають, що алгоритми, які з цих рецензій дізнаються, чи слова на кшталт *феміністичний*, *темношкірий* і *гей* є позитивними чи негативними, можуть засвоїти хибне уявлення від розлючених ботів.

Люди, які використовують ШІ, навчені на створеному людьми тексті, мають очікувати, що разом із ним у систему потрапить певна упередженість, — і планувати, що з нею робити.

Редагування векторів слів прибирає видиме гендерне упередження

Іноді може допомогти невелике редагування. Робін Спір, яка помітила упередження у своєму векторі слів, разом із командою випустила ConceptNet Numberbatch (ні, це не британський актор), де знайшли спосіб відредагувати

гендерне упередження.[14] Спершу команда знайшла спосіб графічно представити вектор слів так, щоб гендерне упередження стало видимим: слова, пов’язані з чоловіками, опинилися ліворуч, а пов’язані з жінками — праворуч.

Після цього, оскільки вони мали одне число, яке показувало силу зв’язку слова з «чоловічим» або «жіночим», то змогли вручну відредагувати це число для певних слів. Результатом став алгоритм, векторні подання слів якого відображають гендерні відмінності, що їх хотіли бачити автори, а не ті, які насправді *були* представлені в інтернеті. Чи розв’язало це редагування проблему упередженості, а чи лише приховало її? Поки що ми не впевнені. До того ж лишається питання, як вирішувати, які слова — якщо такі взагалі є — мають мати гендерні відмінності. Та все ж це краще, ніж дозволити інтернетові вирішувати за нас.

Нейромережа перекручує ім’я Бенедикта Камбербетча

> А ось, без жодної особливої причини, список альтернативних імен Бенедикта Камбербетча, створених нейромережею.

> Bandybat Crumplesnatch
> Bumberbread Calldsnitch
> Butterdink Cumbersand
> Brugberry Cumberront
> Bumblebat Cumplesnap
> Buttersnick Cockersnatch
> Bumbbets Hurmplemon
> Badedew Snomblesoot
> Bendicoot Cocklestink
> Belrandyhite Snagglesnack

Звісно, упередження, яких алгоритми навчаються від нас, не завжди так легко виявити чи відредагувати.

COMPAS однаково «точний», але значно частіше помиляється щодо темношкірих

У 2017 році ProPublica дослідила комерційний алгоритм COMPAS, який широко використовували у США, щоб вирішувати, чи рекомендувати ув’язнених до умовно-дострокового звільнення.[15] Алгоритм ураховував такі чинники, як вік, вид правопорушення й кількість попередніх правопорушень, і на цій основі прогнозував, чи звільнених ув’язнених, імовірно, знову заарештують, чи вчинять вони насильство та/або чи не пропустять наступне судове засідання. Оскільки алгоритм COMPAS був закритою комерційною розробкою, ProPublica могла лише аналізувати ухвалені ним рішення й шукати закономірності.

Виявилося, що COMPAS приблизно у 65 відсотках випадків правильно передбачав повторний арешт підсудного, але його середні оцінки разюче різнилися залежно від раси й статі. Він значно частіше визначав темношкірих підсудних як осіб із високим ризиком, ніж білих, навіть після врахування інших чинників. Унаслідок цього темношкірий підсудний мав значно більшу ймовірність бути помилково позначеним як високоризиковий, ніж білий. У відповідь Northpointe, компанія-продавець COMPAS, зазначила, що її алгоритм має однакову точність для темношкірих і білих підсудних.[16] Проблема в тому, що дані, на яких навчався COMPAS, є результатом сотень років систематичного расового упередження в системі правосуддя США. У США темношкірих значно частіше заарештовують за злочини, ніж білих, хоча злочини вони вчиняють із подібною частотою. Отже, питання, на яке алгоритм в ідеалі мав би відповідати, звучить не «Кого, ймовірно, заарештують?», а «Хто з найбільшою ймовірністю вчинить злочин?». Навіть якщо алгоритм точно передбачає майбутні арешти, він усе одно буде несправедливим, якщо прогнозує расово упереджений рівень арештів.

Домашня адреса дозволяє відновити расу, якої немає в анкеті

Як він узагалі зміг позначати темношкірих підсудних як осіб із високим ризиком арешту, якщо в навчальних даних йому не повідомляли расу? Житлові райони США значною мірою расово сегреговані, тож він міг вивести расу лише з домашньої адреси підсудного. Можливо, він помітив, що людям із певного району рідше дають умовно-дострокове звільнення або їх частіше заарештовують, і відповідно сформував своє рішення.

«Альтернативні дані» дають страховому ШІ таємний шлях до дискримінації

ШІ настільки схильні знаходити й використовувати людські упередження, що штат Нью-Йорк нещодавно оприлюднив настанови для страхових компаній: якщо вони аналізують «альтернативні дані», здатні підказати ШІ, у якому районі живе людина, то можуть порушувати антидискримінаційні закони. Законодавці визнали, що це був би підступний обхідний шлях, який дозволить ШІ визначити ймовірну расу людини, а потім шахрайськи досягти людського рівня результативності, застосувавши расизм або інші форми дискримінації.[17]

Зрештою, передбачити, які злочини чи нещасні випадки можуть статися, — дуже складна й широка задача. Виявити й скопіювати упередження для ШІ значно простіше.

ЦЕ НЕ РЕКОМЕНДАЦІЯ — ЦЕ ПРОГНОЗ

Алгоритм Amazon карає резюме за слово «жіночий»

ШІ дають нам саме те, про що ми просимо, тому треба дуже обережно обирати свої прохання. Розгляньмо, наприклад, відбір кандидатів на роботу. У 2018 році Reuters повідомив, що Amazon відмовилася від інструмента, який випробовувала для попереднього відбору претендентів, після того як тести компанії показали: ШІ дискримінує жінок. Він навчився знижувати оцінку резюме кандидаток, які навчалися в жіночих закладах освіти, і навіть карав резюме, де згадувалося англійське слово *women's* («жіночий») — наприклад, у фразі *women's soccer team* («жіноча футбольна команда»).[18] На щастя, компанія виявила проблему до того, як ці алгоритми почали використовувати для реальних рішень про відбір.[19] Програмісти Amazon не прагнули створити упереджений алгоритм — то як він вирішив віддавати перевагу чоловікам?

Якщо алгоритм навчають на тому, як менеджери з найму раніше відбирали чи ранжували резюме, він із великою ймовірністю засвоїть упередження. Добре задокументовано, що люди перевіряють резюме із сильним гендерним і расовим упередженням — навіть якщо це роблять жінки та/або представники меншин та/або люди, які не вважають себе упередженими. Резюме, подане з чоловічим ім'ям, має значно більші шанси привести до співбесіди, ніж ідентичне резюме з жіночим ім'ям. Навчання алгоритму віддавати перевагу резюме, схожим на резюме найуспішніших працівників компанії, також може обернутися проти задуму, якщо її персоналові вже бракує різноманіття або якщо компанія нічого не зробила з гендерним упередженням у своїх оцінках результативності.[20]

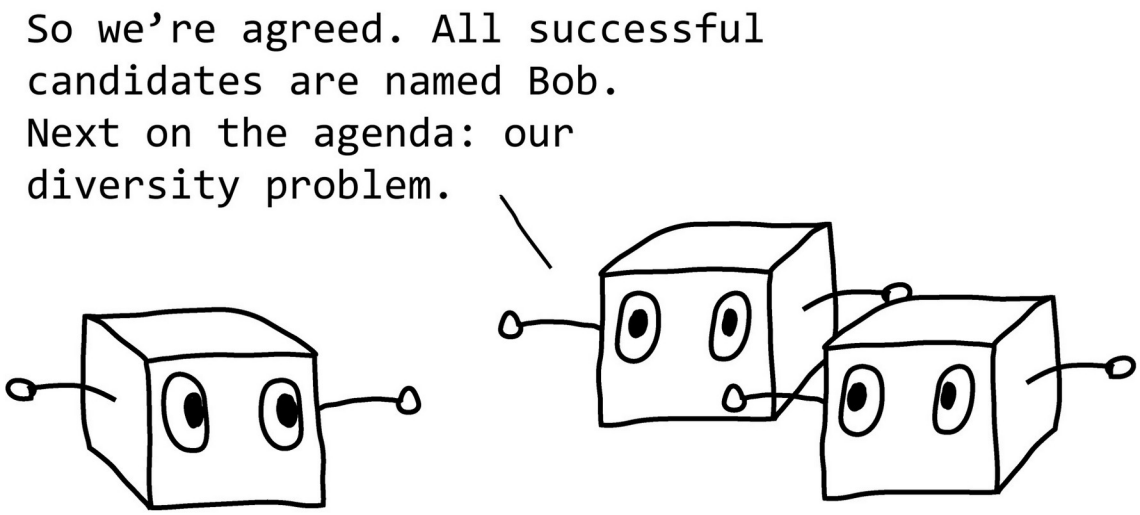
Ідеальний кандидат зветься Джаредом і грає в лакрос

> В інтерв'ю *Quartz* Марк Дж. Жіруар, юрист із трудового права в міннеаполіській фірмі Nilan Johnson Lewis, розповів про клієнта, який перевіряв алгоритм найму іншої компанії й хотів з'ясувати, які ознаки алгоритм найсильніше пов'язував із високою результативністю. Цими ознаками були: (1) кандидата звали Джаред і (2) кандидат грав у лакрос.[21]

Навіть після видалення жіночих слів алгоритм знаходить чоловічі

Коли інженери Amazon виявили упередженість свого інструмента перевірки резюме, вони спробували її усунути, видаливши слова, пов'язані з жінками, з переліку тих, які алгоритм брав до уваги. Їхню роботу ще більше ускладнювало те, що алгоритм також навчався віддавати перевагу словам, які найчастіше трапляються в чоловічих резюме, — наприклад, *executed* («виконав») і *captured* («захопив»). Виявилося, що алгоритм чудово відрізняє

чоловічі резюме від жіночих, але в усьому іншому жахливо рекомендує кандидатів, повертаючи фактично випадкові результати. Зрештою Amazon закрила проєкт.



Алгоритм не радить найкраще рішення — він передбачає людську поведінку

Люди поводяться з такими алгоритмами так, наче ті дають рекомендації, але значно точніше сказати, що вони роблять прогнози. Вони не повідомляють нам, яке рішення було б найкращим, а лише навчаються передбачати людську поведінку. Оскільки люди схильні до упереджень, алгоритми, які від них навчаються, також будуть упередженими, якщо люди не докладуть додаткових зусиль, щоб знайти й усунути ці упередження.

Прогнозна поліція спрямовує більше патрулів туди, де вже більше арештів

Застосовуючи ШІ для розв'язання реальних проблем, ми також маємо уважно дивитися, *що саме* прогнозується. Існує різновид алгоритмів, який називають **прогнозною поліцейською діяльністю**: він аналізує минулі поліцейські записи й намагається передбачити, де й коли в майбутньому будуть зареєстровані злочини. Коли поліція бачить, що алгоритм спрогнозував злочин у певному районі, вона може відправити туди більше працівників, щоб спробувати запобігти злочину або принаймні опинитися поруч, коли він станеться. Однак алгоритм передбачає не місце, де станеться найбільше злочинів, а те, де найбільше злочинів буде *виявлено*. Якщо в певний район відправити більше поліції, там виявлять більше злочинів, ніж у районі, який мало патрулюють, але який так само потерпає від злочинності, — просто тому, що поруч більше поліціянтів, здатних побачити інциденти й зупиняти

випадкових людей. А коли рівень виявленої злочинності в районі зросте, поліція може вирішити направити туди ще більше працівників. Цю проблему називають надмірним патрулюванням, і вона може створити петлю зворотного зв'язку, у якій повідомляють про дедалі вищий рівень злочинності. Проблема посилюється, якщо в реєстрації злочинів є расове упередження: якщо поліція схильна частіше зупиняти чи заарештовувати людей певної раси, їхні райони можуть зрештою опинитися під надмірним поліцейським наглядом. Додайте до цього алгоритм прогнозної поліцейської діяльності — й проблема може лише погіршитися, особливо якщо ШІ навчали на даних поліцейських управлінь, які, наприклад, підкидали наркотики невинним людям, щоб виконати план арештів.[22]

ПЕРЕВІРЯЙТЕ ЇХНЮ РОБОТУ

Математичне відбілювання ховає людське упередження всередині ШІ

Як не дати ШІ ненавмисно копіювати людські упередження? Одна з головних речей, які ми можемо зробити, — очікувати, що це станеться. Не слід вважати рішення ШІ справедливими лише тому, що ШІ не здатний затаїти образу. Практику вважати рішення неупередженим лише тому, що воно походить від ШІ, іноді називають **математичним відбілюванням** або **відмиванням упередженості**. Упередження нікуди не зникло, адже ШІ скопіював його зі своїх навчальних даних, але тепер воно загорнуте в шар важкої для тлумачення поведінки ШІ. Навмисно чи ні, компанії можуть зрештою використовувати ШІ, який дискримінує в украй незаконні, але, можливо, прибуткові способи.

Тому нам треба перевіряти ШІ, щоб переконатися, що їхні кмітливі рішення не жахливі.

Сушарки, голосові системи й розпізнавання облич провалюють пізні тести

Один із найпоширеніших способів виявлення проблем — піддати алгоритм суворим випробуванням. На жаль, іноді ці тести проводять уже після того, як алгоритм почали використовувати: наприклад, коли користувачі помічають, що сушарки для рук не реагують на темну шкіру, розпізнавання голосу менш точне для жінок, ніж для чоловіків, або що три провідні алгоритми розпізнавання облич значно гірше розпізнають темношкірих жінок, ніж світлошкірих чоловіків.[23] У 2015 році дослідники з Університету Карнеґі — Меллона використали інструмент AdFisher для аналізу оголошень про роботу в

Google і виявили: ШІ значно частіше рекомендував високооплачувані керівні посади чоловікам, ніж жінкам.[24] Можливо, роботодавці саме цього й просили, а можливо, ШІ випадково навчився так діяти без відома Google.

Це найгірший сценарій: проблему виявляють уже після того, як вона заподіяла шкоду.

Різноманітні команди краще помічають приховану дискримінацію

В ідеалі такі проблеми варто передбачати й проєктувати алгоритми так, щоб вони взагалі не виникали. Як? По-перше, потрібна різноманітніша робоча сила в технологічній сфері. Програмісти, які самі належать до маргіналізованих груп, із більшою ймовірністю передбачать, де в навчальних даних може ховатися упередження, і серйозніше поставляться до таких проблем (також допомагає, якщо цим працівникам дають повноваження вносити зміни). Звісно, це не усуне всіх проблем. Навіть програмістів, які знають, як алгоритми машинного навчання можуть поводитися неправильно, ті все одно регулярно дивують.

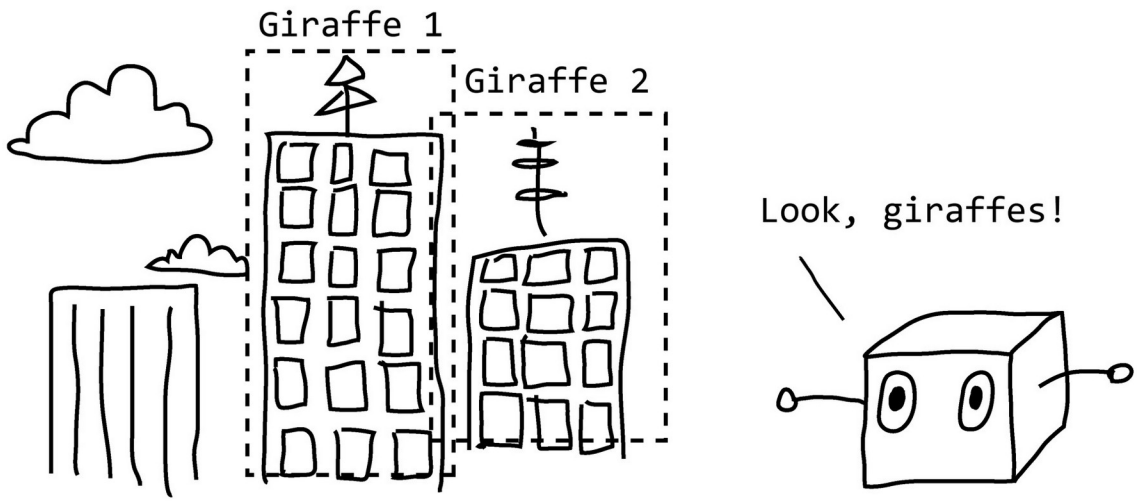
Themis видає половину позик жінкам — але всі вони з однієї країни

Тому також важливо суворо перевіряти алгоритми, перш ніж випускати їх у світ. Люди вже створили програмне забезпечення, яке систематично шукає упередження в програмах, що, наприклад, вирішують, чи схвалити певному заявникові позику.[25] У цьому прикладі програма для перевірки упереджень систематично випробовувала б безліч вигаданих претендентів на позику, шукаючи закономірності в характеристиках тих, кого схвалили. Такий потужний систематичний підхід найкорисніший, бо прояви упередження іноді бувають дивними. Одна програма перевірки упереджень під назвою Themis шукала гендерну упередженість у заявках на позику. Спершу все виглядало добре: приблизно половину позик отримували чоловіки, а половину — жінки (даних про інші гендери не повідомляли). Але коли дослідники вивчили географічний розподіл, то виявили, що упередженості все одно багато: 100 відсотків жінок, які отримали позики, були з однієї країни. Уже з’явилися компанії, які пропонують перевірку на упередженість як послугу.[26] Якщо уряди й галузі почнуть вимагати сертифікації нових алгоритмів щодо упередженості, ця практика може стати значно поширенішою.

Пояснюваний ШІ показує, на які пікселі та ознаки дивився

Ще один спосіб, яким люди виявляють упередження та іншу небажану поведінку, — створення алгоритмів, здатних пояснити, як вони дійшли своїх

рішень. Це складно, бо, як ми вже бачили, людям загалом важко тлумачити ШІ. І, як ми знаємо з розглянутого в розділі 4 візуального чатбота, алгоритм важко навчити осмислено відповідати на запитання про те, як він бачить світ. Найбільшого поступу досягли з алгоритмами розпізнавання зображень, які можуть указати на фрагменти зображення, яким приділяли увагу, або показати різновиди ознак, які шукали.



Кілька підалгоритмів можуть звітувати про зрозумілі людині рішення

Також може допомогти побудова алгоритмів із групи підалгоритмів, якщо кожен підалгоритм повідомлятиме рішення у зрозумілій людині формі.

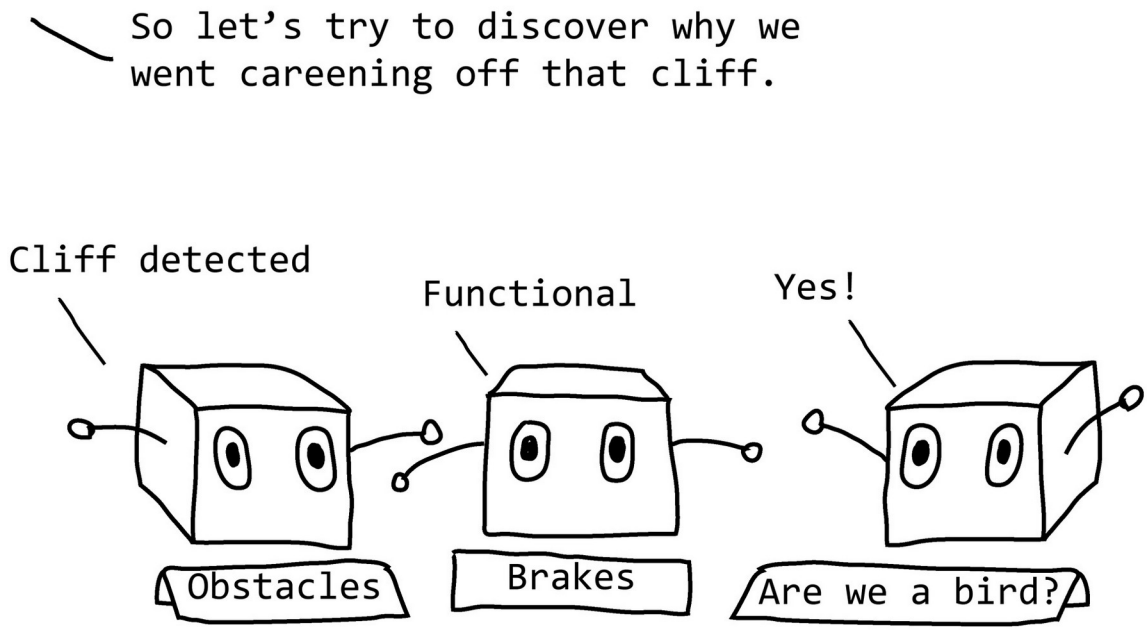
Попередня обробка змінює навчальні рішення, доки упередження не зникне

Що можна зробити, коли ми виявили упередження? Один зі способів вилучити його з алгоритму — редагувати навчальні дані доти, доки в них більше не буде упередження, яке нас турбує.[27] Наприклад, ми можемо перенести деякі заявки на позику з категорії «відхилено» до категорії «схвалено» або вибірково взагалі вилучити певні заявки з навчальних даних. Це називають **попередньою обробкою**.

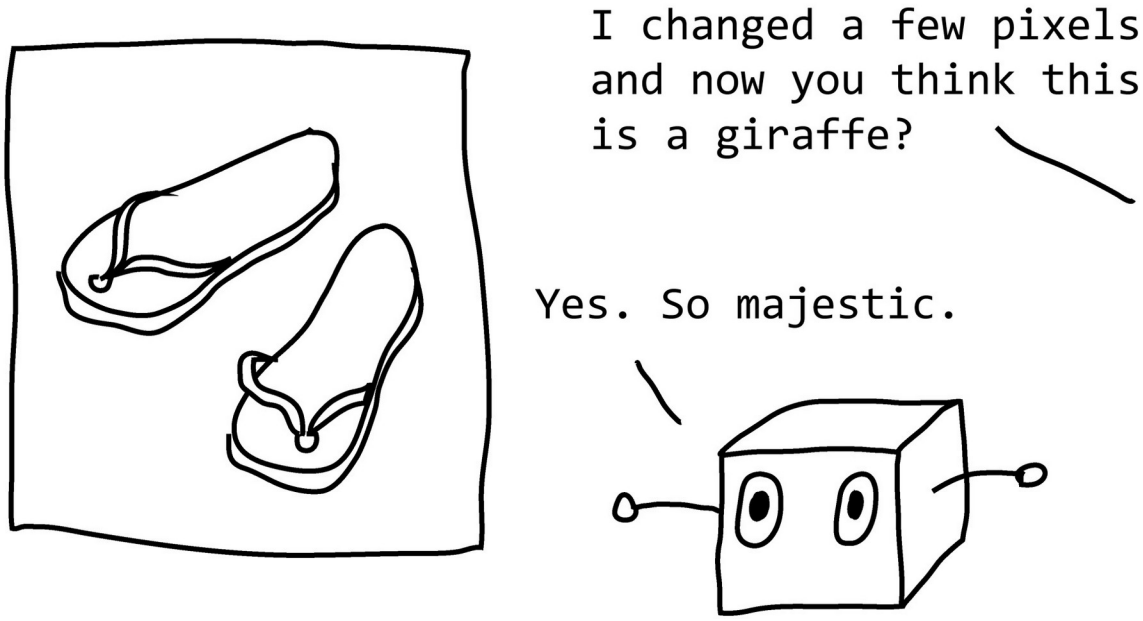
Людина має перевірити, чи «геніальне рішення» ШІ не змушує хапатися за голову

Ключем до всього цього може бути людський нагляд. Оскільки ШІ так схильні несвідомо розв’язувати не ту задачу, ламати речі чи обирати невдалі скорочення шляху, нам потрібні люди, які перевірятимуть, чи їхнє «геніальне рішення» не змушує хапатися за голову. А ці люди мають добре знати способи, якими ШІ зазвичай досягають успіху або помиляються. Це трохи схоже на

перевірку роботи колеги — дуже, дуже дивного колеги. Щоб побачити, наскільки саме дивного, у наступному розділі ми розглянемо кілька способів, якими ШІ схожий на людський мозок, і кілька способів, якими він від нього дуже відрізняється.



РОЗДІЛ 8. Чи схожий мозок ШІ на людський?



Комп’ютерний код незалежно відкриває явища живого мозку

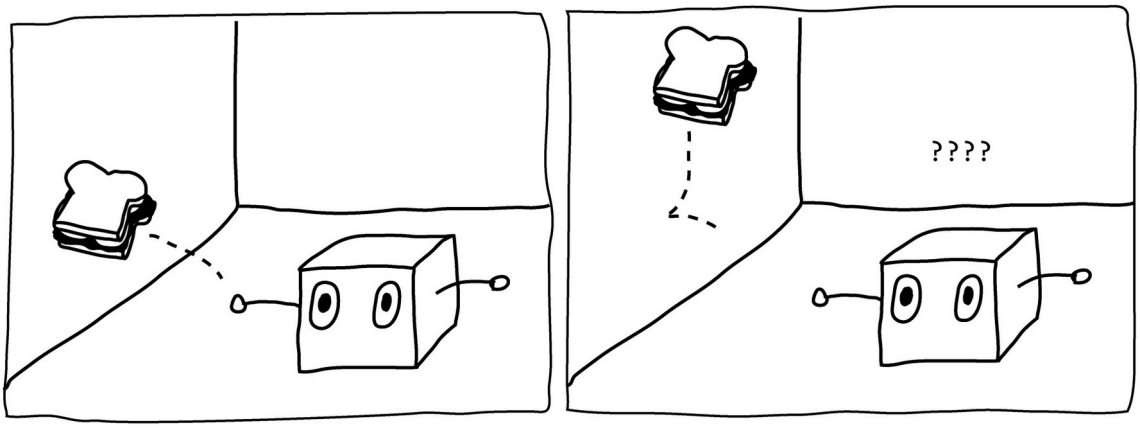
Алгоритми машинного навчання — лише рядки комп’ютерного коду, але, як ми вже бачили, вони можуть робити речі, що здаються дуже людськими: навчатися, випробовуючи стратегії, обирати ліниві короткі шляхи для розв’язання задач або взагалі уникати перевірки, видаляючи відповіді. До того ж будова багатьох алгоритмів машинного навчання натхненна прикладами з реального життя. Як ми дізналися з розділу 3, нейронні мережі дуже приблизно наслідують нейрони людського мозку, а еволюційні алгоритми — біологічну еволюцію. Виявляється, багато явищ, що виникають у мозку чи живих організмах, також з’являються у ШІ, які їх наслідують. Іноді вони навіть виникають незалежно, без навмисного програмування з боку людини.

СВІТИ СНІВ ШІ

Ви здатні уявити політ сендвіча, бо маєте внутрішню модель світу

Уявіть, що з усієї сили кидаєте сендвіч у стіну. (Якщо так легше, уявіть один із жахливих відхилених сендвічів із розділу 3.) Зосередившись, ви, ймовірно,

зможете яскраво уявити кожен етап: гладенькі чи горбкуваті скибки хліба між пальцями; текстуру скоринки, якщо жбурляєте багет або булочку. Ви, мабуть, уявите, наскільки хліб піддасться під пальцями: можливо, вони трохи втиснуться, але не пройдуть наскрізь. Ви також можете уявити траєкторію руки, коли відводите її назад для кидка, і точку руху, у якій випустите сендвіч. Ви знаєте, що він вилетить із руки за інерцією й може трохи хитатися чи обертатися в повітрі. Ви навіть здатні передбачити, де й з якою силою він удариться об стіну, як хліб може деформуватися чи розірватися і що станеться з начинкою. Ви знаєте, що сендвіч не здійметься, наче повітряна кулька, не зникне й не заблимає зеленим та помаранчевим. (Ну, хіба що це сендвіч з арахісовим маслом, гелієм та інопланетним артефактом.)



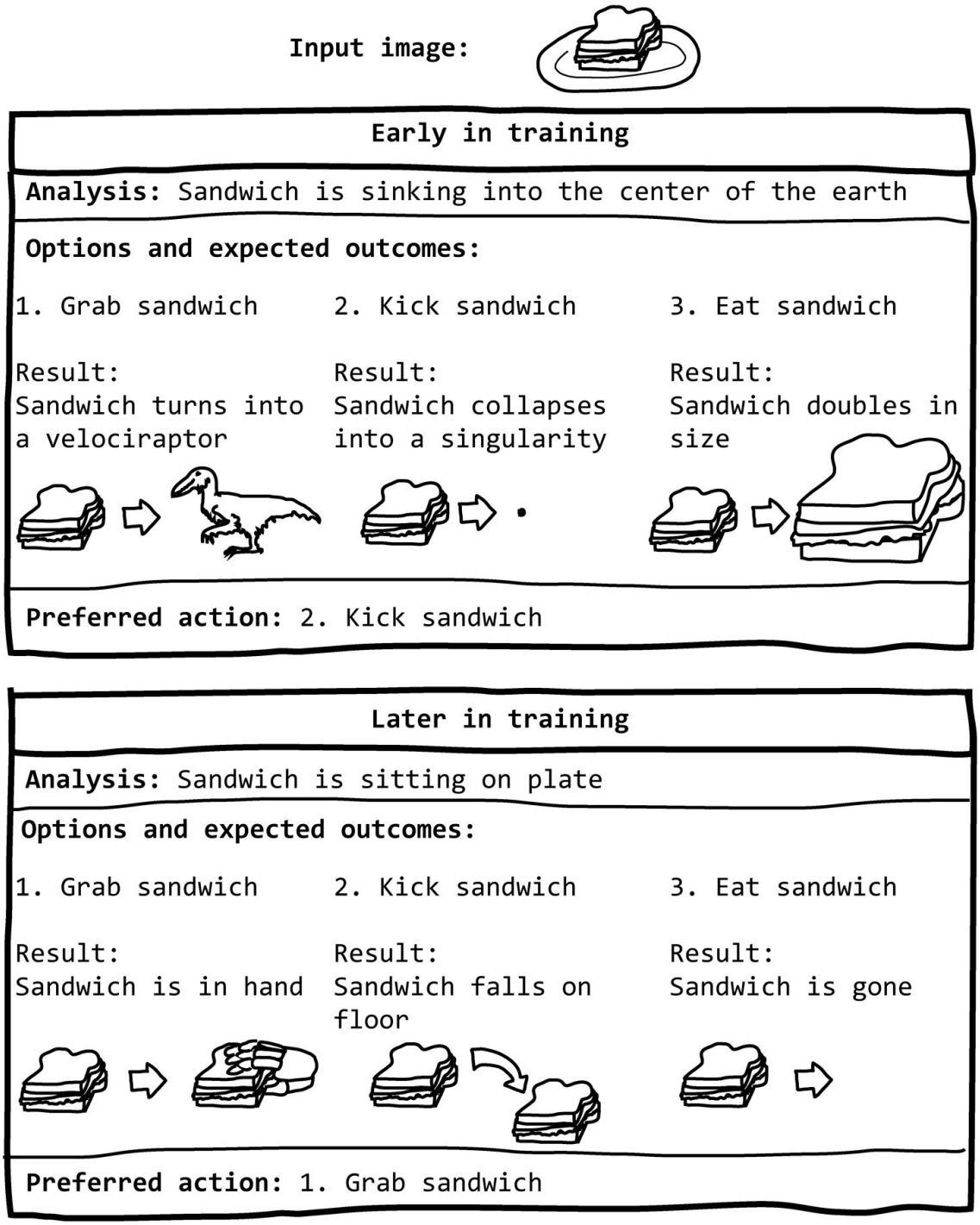
Внутрішня модель дозволяє відбити м’яч швидше за нервовий сигнал

Коротко кажучи, у вас є внутрішні моделі сендвічів, фізики кидання предметів і стін. Нейробіологи досліджували ці внутрішні моделі, які керують нашим сприйняттям світу й прогнозами щодо майбутнього. Коли бейсболіст замахується битою на м’яч, його руки починають рухатися задовго до того, як м’яч залишить руку пітчера: м’яч навіть не перебуває в повітрі достатньо довго, щоб нервові імпульси встигли дістатися м’язів гравця. Замість оцінювати політ м’яча, бейсболіст покладається на внутрішню модель поведінки подачі, щоб правильно розрахувати час замаху. Багато наших найшвидших рефлексів працюють так само, використовуючи внутрішні моделі для прогнозування найкращої реакції.

Одна частина ШІ спостерігає, друга прогнозує, третя обирає дію

Люди, які створюють ШІ для навігації реальними або змодельованими місцевостями чи для розв’язання інших задач, також часто надають їм внутрішні моделі. Одну частину ШІ можна спроєктувати так, щоб вона

спостерігала за світом, виділяла важливі фрагменти інформації й використовувала їх для побудови або оновлення внутрішньої моделі. Інша частина використовуватиме модель, щоб передбачити наслідки різних дій. Ще одна частина вирішуватиме, який результат найкращий. У процесі навчання ШІ вдосконалюється в усіх трьох завданнях. Люди навчаються дуже схожим способом — постійно формуючи й оновлюючи припущення про навколишній світ.



Сни дають мозку й ШІ безпечний тренувальний майданчик

Деякі нейробіологи вважають, що сновидіння — це спосіб використовувати внутрішні моделі для тренувань із низькими ставками. Хочете випробувати сценарії втечі від розлюченого носорога? Значно безпечніше перевірити їх уві сні, ніж дражнити справжнього носорога. Спираючись на цей принцип, програмісти машинного навчання інколи використовують тренування у снах,

щоб допомогти алгоритмам навчатися швидше. У розділі 3 ми розглядали алгоритм — насправді три ШІ в одному, — метою якого було якомога довше лишатися живим на одному рівні комп'ютерної гри Doom.[1] Поєднавши візуальне сприйняття екрана гри, пам'ять про минулі події й прогноз наступної події, програмісти побудували алгоритм, здатний створювати внутрішню модель рівня та використовувати її для вибору дій. Як і у прикладі з бейсболістом, внутрішні моделі — один із найкращих інструментів для навчання алгоритмів діяти.

У сні Doom вогняні кулі важливіші за самих монстрів

Особливий поворот полягав у тому, що ШІ тренувався не в реальній грі, а всередині самої моделі: він випробовував нові стратегії у власній сновидній версії гри, а не в справжній. Це має певні переваги: оскільки ШІ здебільшого навчився будувати модель із найважливіших деталей, сновидна версія потребує менше обчислень. Процес також прискорює навчання, бо ШІ може зосередитися на важливих деталях та ігнорувати решту. На відміну від людських снів, сни ШІ дозволяють нам зазирнути у внутрішню модель, наче ми підглядаємо його сновидіння. Ми бачимо схематичну, розмиту версію ігрового рівня. Наскільки важливою ШІ вважає кожну особливість гри, можна оцінити за детальністю, з якою її відтворено у світі сну. Тут монстри, що кидають вогняні кулі, лише ледь окреслені, зате самі кулі промальовані реалістично. Цікаво, що цегляні візерунки на стінах теж присутні у внутрішній моделі: можливо, вони важливі для оцінювання відстані гравця до стіни.

І справді, у цій спрощеній версії всесвіту ШІ може відточувати навички прогнозування та ухвалення рішень, зрештою навчившись уникати більшості вогняних куль. Навички, здобуті у світі сну, переносяться й у справжню комп'ютерну гру, тож, тренуючись у внутрішній моделі, ШІ краще грає в оригінал.

ШІ зламує власний сон, щоб монстри перестали стріляти

Однак не всі випробувані уві сні стратегії ШІ працювали в реальному світі. Зокрема, він навчився зламувати власний сон — так само як усі ті ШІ з розділу 6, що зламували свої симуляції. Рухаючись певним чином, ШІ виявив, що може використати збій внутрішньої моделі, який не давав монстрам узагалі випускати вогняні кулі. У справжньому світі ця стратегія, звісно, не спрацювала. Люди-сновидці часом так само розчаровуються, коли прокидаються й виявляють, що більше не вміють літати.

СПРАВЖНІ Й ШТУЧНІ МОЗКИ ДУМАЮТЬ ПОДІБНО

ШІ, який грав у Doom, мав внутрішню модель світу, бо програмісти свідомо надали її йому. Але бувають випадки, коли нейронні мережі незалежно приходять до тих самих стратегій, які нейробіологи відкрили в мозку тварин.

Без учителя нейромережа самостійно відкриває фільтри країв

У 1997 році дослідники Ентоні Белл і Терренс Сейновскі навчили нейронну мережу розглядати різні природні сцени («дерева, листя й таке інше») та з’ясовувати, які ознаки вона здатна виявляти. Ніхто не казав їй, що саме шукати, — лише відокремлювати речі, які відрізняються. (Такий вільний аналіз набору даних називають **навчанням без учителя**.) Мережа зрештою самостійно виробила низку фільтрів для виявлення країв і візерунків, подібних до тих, які науковці виявили в зорових системах людей та інших ссавців. Без спеціальної вказівки штучна нейронна мережа прийшла до деяких тих самих прийомів обробки зображень, якими користуються тварини.[2]

Навігаційний ШІ незалежно створює аналоги ґратчастих клітин

Були й інші подібні випадки. Дослідники Google DeepMind виявили, що алгоритми, створені для навчання навігації, самостійно розвивали представлення ґратчастих клітин, схожі на ті, що є в мозку деяких ссавців.[3]

Видалення окремих штучних нейронів прибирає дерева й збої

У певному сенсі на нейронних мережах працює навіть операція на мозку. Згадайте: у розділі 3 я описувала, як дослідники вивчали нейрони мережі для генерування зображень — GAN — і змогли визначити окремі нейрони, що створювали дерева, куполи, цеглу та вежі. Вони також знайшли нейрони, які, схоже, породжували глючні плями. Коли ці нейрони вилучили з мережі, збої зникли із зображень. Дослідники також з’ясували, що можуть вимикати нейрони, які генерували певні об’єкти, — і ці об’єкти справді зникали із зображень.[4]

Цифрова й біологічна еволюція винаходять ті самі моторошні стратегії

> **КОНВЕРГЕНТНА ЕВОЛЮЦІЯ**
>
> Віртуальні нервові системи — не єдині речі, здатні нагадувати свої реальні відповідники. Цифрові версії еволюції можуть виробляти поведінку, що також еволюціонувала у справжніх організмів: співпрацю, конкуренцію, обман, хижацтво й навіть паразитизм. Навіть деякі з найдивніших стратегій цифрово еволюціонованих ШІ мають відповідники в реальному житті.

> На одній віртуальній арені під назвою PolyWorld, де змодельовані організми змагалися за їжу й ресурси, деякі істоти виробили доволі похмуру стратегію — поїдати власних дітей. Створення дітей у тому світі не витрачало ресурсів, а самі діти були безкоштовним джерелом їжі.[5] І так, справжні організми також еволюціонували до певної версії цієї поведінки. Деякі комахи, земноводні, риби й павуки відкладають незапліднені **трофічні яйця** спеціально для того, щоб їх з'їдало потомство. Іноді ці яйця є додатковою їжею, а в інших випадках — як у рийного клопа *Canthophorus niveimarginatus* — молодняк цілком залежить від них як джерела харчування.[6] Деякі мурахи й бджоли навіть виробляють трофічні яйця як їжу для своїх маток. І не лише яйця з'їдають їхні брати й сестри. Деякі акули народжують живих дитинчат, і ті, що доживають до народження, вижили, поїдаючи своїх братів і сестер ще в утробі.

КАТАСТРОФІЧНЕ ЗАБУВАННЯ

Друге вузьке завдання стирає з пам'яті перше

Згадайте з розділу 2: що вужче завдання ШІ, то розумнішим він здається. Не можна взяти вузький штучний інтелект, навчати його виконувати завдання за завданням і зрештою отримати загальний штучний інтелект. Якщо спробувати навчити вузький ШІ другого завдання, він забуде перше. У результаті ви матимете вузький ШІ, який навчився лише того, чого ви навчали його останнім.

Я постійно бачу це на практиці, коли навчаю нейронні мережі генерувати текст.

Мережа вигадує правдоподібні закляття Dungeons & Dragons

Наприклад, ось частина результатів нейромережі, яку я навчала на великій кількості заклять Dungeons & Dragons. Вона виконала завдання доволі добре: ці закляття можна вимовити, вони звучать правдоподібно й навіть можуть переконати людей, що справжні. (Щоправда, я відібрала найкращі результати.)

> Find Faithful («Знайти вірного»)
> Entangling Stone («Камінь обплутування»)
> Bestow Missiles («Дарувати снаряди»)
> Energy Secret («Таємниця енергії»)
> Resonating Mass («Резонансна маса»)
> Mineral Control Spell («Закляття керування мінералами»)
> Holy Ship («Святий корабель»)
> Night Water («Нічна вода»)
> Feather Fail («Провал пера»)

> Hail to the Dave («Слава Дейву»)
> Delay Tail («Відкласти хвіст»)
> Stunker’s Crack («Тріщина смердючника»)
> Combustive Blaps («Горючі блапси»)
> Blade of the Darkstone («Клинок темного каменя»)
> Distracting Sphere («Сфера відволікання»)
> Love Hatter («Любовний капелюшник»)
> Seed of Dance («Насіння танцю»)
> Protection of Person of Ability («Захист здібної особи»)
> Undead Snow («Сніг нежиті»)
> Curse of King of Furch («Прокляття короля Ферча»)

Після навчання на пирогах закляття стають кремовими й хрусткими

Потім я навчила ту саму нейронну мережу на новому наборі даних — назвах рецептів пирогів. Чи отримала б я нейромережу, здатну створювати і пироги, і закляття? Після зовсім нетривалого навчання здавалося, що саме це й починає відбуватися: закляття D&D набули виразного присмаку.

> Discern Pie («Розпізнати пиріг»)
> Detect Cream («Виявити крем»)
> Tart of Death («Тарт смерті»)
> Summon Fail Pie («Викликати пиріг провалу»)
> Death Cream Swarm («Рій смертельного крему»)
> Easy Apple Cream Tools («Легкі яблучно-кремові інструменти»)
> Bear Sphere Transport Pie («Пиріг перенесення ведмежої сфери»)
> Crust Hammer («Молот скоринки»)
> Glow Cream Pie («Пиріг сяйливого крему»)
> Switch Minor Pie («Перемкнути малий пиріг»)
> Wall of Tart («Стіна тарту»)
> Bomb Cream Pie («Пиріг бомбового крему»)
> Crust Music («Музика скоринки»)
> Arcane Chocolation («Таємнича шоколадація»)
> Tart of Nature («Тарт природи»)
> Mordenkainen’s Pie («Пиріг Морденкайнена»)
> Rary’s Or Tentacle Cheese Cruster («Сироскоринник Рарі або щупальця»)
> Haunting Pie («Пиріг із привидами»)
> Necroppostic Crostility («Некропостична скоринність»)
> Tartle of the Flying Energy Crum («Тартль летючої енергетичної крихти»)

Зрештою мережа чудово називає пироги, але перестає бути чарівником

На жаль, із продовженням навчання нейромережа швидко почала забувати засвоєні закляття. Вона стала добре генерувати назви пирогів. Насправді вона стала генерувати їх *чудово*. Але чарівником більше не була.

> Baked Cream Puff Cake («Печений торт із кремовими профітролями»)
> Reese's Pecan Pie («Пекановий пиріг Reese's»)
> Eggnog Peach Pie #2 («Персиковий пиріг з еґноґом № 2»)
> Apple Pie With Fudge Treats («Яблучний пиріг із ласощами з фаджу»)
> Almond-Blackberry Filling («Мигдально-ожинова начинка»)
> Marshmallow Squash Pie («Гарбузовий пиріг із маршмелоу»)
> Cromberry Yas («Кромберрі яс»)
> Sweet Potato Piee («Пирігі з батату»)
> Cheesy Cherry Cheese Pie #2 («Сирний вишнево-сирний пиріг № 2»)
> Ginger Impossible Strawberry Tart («Імбирний неможливий полуничний тарт»)
> Coffee Cheese Pie («Кавово-сирний пиріг»)
> Florid Pumpkin Pie («Пишномовний гарбузовий пиріг»)
> Meat-de-Topping («М'ясо-де-посипка»)
> Baked Trance Pie («Печений трансовий пиріг»)
> Fried Cream Pies («Смажені кремові пироги»)
> Parades Or Meat Pies Or Cake #1 («Паради, або м'ясні пироги, або торт № 1»)
> Milk Harvest Apple Pie («Яблучний пиріг молочного врожаю»)
> Ice Finger Sugar Pie («Цукровий пиріг крижаного пальця»)
> Pumpkin Pie With Cheddar Cookie («Гарбузовий пиріг із чеддеровим печивом»)
> Fish Strawberry Pie («Рибно-полуничний пиріг»)
> Butterscotch Bean Pie («Квасолевий пиріг із тягучою карамеллю»)
> Caribou Meringue Pie («Пиріг із карибу та меренгою»)

Нові зв'язки перепрофільовують усі нейрони й стирають давню навичку

Цю особливість нейронних мереж називають **катастрофічним забуванням**.[7] Типова нейронна мережа ніяк не захищає довготривалу пам'ять. Коли вона вивчає нові завдання, усі її нейрони можна перепризначити: зв'язки, що слугували написанню заклять, перебудовуються й натомість використовуються для вигадування пирогів. Катастрофічне забування — один із чинників, що визначають, які проблеми практично розв'язувати за допомогою сучасних ШІ, і воно впливає на наше уявлення про те, як змушувати ШІ щось робити.

Дослідники працюють над подоланням катастрофічного забування. Зокрема, намагаються вбудувати своєрідну довготривалу пам'ять із захищених

нейронів — подібно до того, як людський мозок надійно зберігає довготривалі спогади десятиліттями.

GPT-2 пам’ятає Гаррі Поттера, але зводить розмову до супу

Більші нейронні мережі можуть бути трохи стійкішими до катастрофічного забування, можливо, тому що їхні здібності розподілені між такою кількістю навчених клітин, що під час перенесення навчання перепризначаються не всі. Великий алгоритм на кшталт GPT-2 — великої текстової нейромережі з розділу 2 — усе ще здатний генерувати фанфіки про Гаррі Поттера навіть після того, як я довго навчала його на рецептах. Мені досить дати йому початок історії про Гаррі та Снейпа, і навчений на рецептах GPT-2 згадує, як дописувати решту. Кумедно, що він схильний спрямовувати історію до розмов про їжу. Дайте йому абзац із роману жахів — і зрештою персонаж почне ділитися рецептами та згадувати «сендвіч у шоколаді з маслом і сиром»; а розмова Люка Скайвокера з Обі-Ваном Кенобі незабаром перетвориться на обговорення альдераанського рибного соусу. Усього за кілька абзаців історія, що починалася зі звинувачення Гаррі Снейпом у викраденні зіль, стала ось цією застільною розмовою про поліпшення рецепта супу.

> «Але мені все ж цікаво, чи ви справді їли цей суп із невеликою кількістю риби. Суп настільки сповнений смаку, що там не було навіть жодного смаку».
>
> «Ми з’їли його з цілою купою цього», — зауважила Герміона. «Ми всі їмо це з рибою. Мабуть, воно доволі смачне».
>
> «Гадаю, так», — погодився Гаррі. «Я куштував його зі смаженими устрицями, з омаром, із креветками й на хвостах омара. Він дуже добрий».
>
> «Гадаю, насправді це був лише рецепт смажених устриць».
>
> «Що це було?» — сказав Рон із кухні.
>
> «Для мене це дуже особливий суп, бо він такий незвичайний. Треба почати зі смаку, а потім поступово додавати інші інгредієнти».

Навіть якщо ШІ стане досить великим, щоб виконувати кілька тісно пов’язаних завдань одночасно, він може зрештою виконувати кожне з них дещо погано. Згадайте нейромережу для генерування котів із розділу 4, яка насилу давала раду різним позам котів.

Найпростіше рішення — окремий вузький ШІ для кожного завдання

Поки що найпоширенішим розв’язанням катастрофічного забування є поділ на модулі: щоразу, коли ми хочемо додати нове завдання, використовуємо новий ШІ. У результаті маємо кілька незалежних ШІ, кожен із яких уміє робити лише одну річ. Але якщо з’єднати їх усіх і придумати спосіб визначати, який ШІ

потрібен у кожну мить, формально ми отримаємо алгоритм, здатний робити більше ніж одну річ. Згадайте ШІ для Doom, який насправді складався з трьох ШІ: один спостерігав за світом, другий передбачав наступну подію, а третій обирав найкращу дію.

Майбутній загальний ШІ може нагадувати рій мурах, а не людину

Деякі дослідники вважають катастрофічне забування однією з головних перешкод на шляху до інтелекту людського рівня. Якщо алгоритм може вивчати лише одне завдання за раз, як він упорається з величезним різноманіттям розмовних, аналітичних, планувальних і пов’язаних з ухваленням рішень завдань, які виконують люди? Можливо, катастрофічне забування завжди обмежуватиме нас алгоритмами для одного завдання. З іншого боку, якби достатня кількість однозадачних алгоритмів могла координуватися, наче мурахи чи терміти, вони розв’язували б складні проблеми, взаємодіючи між собою. Майбутні загальні штучні інтелекти, якщо вони існуватимуть, можуть бути більше схожими на рій суспільних комах, ніж на людей.

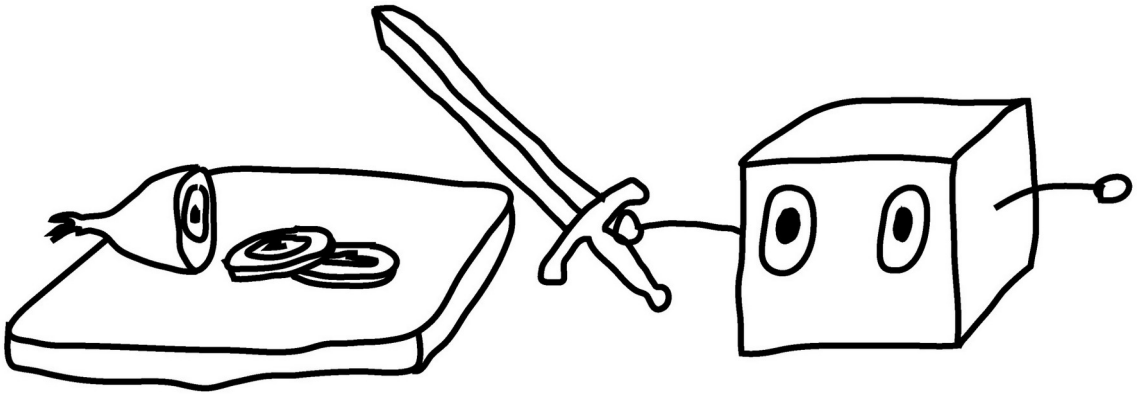
ПОСИЛЕННЯ УПЕРЕДЖЕНЬ

У розділі 7 ми побачили деякі з численних способів, якими ШІ можуть засвоювати упередження зі своїх навчальних даних. Далі стає лише гірше.

ШІ не просто копіює упередження, а робить його сильнішим

Алгоритми машинного навчання не лише підхоплюють упередження з навчальних даних, а й схильні ставати *ще* упередженішими за ці дані. З їхнього погляду, вони лише відкрили корисне спрощене правило, яке допомагає частіше збігатися з людьми у своїх навчальних даних.

Легко зрозуміти, чому спрощені правила можуть бути корисними. Алгоритм розпізнавання зображень може погано розпізнавати предмети в руках, але якщо він також бачить кухонну стільницю, шафки й плиту, то може припустити, що людина на знімку тримає кухонний ніж, а не меч. Насправді навіть якщо він гадки не має, як відрізнити меч від кухонного ножа, це неважливо, доки він знає, що на кухні здебільшого слід відповідати «кухонний ніж». Це приклад проблеми дисбалансу класів із розділу 6, коли алгоритм класифікації бачить значно більше прикладів одного виду вхідних даних, ніж іншого, і засвоює: можна безкоштовно отримати чималу точність, якщо припустити, що рідкісні випадки ніколи не трапляються.



33 відсотки чоловіків на кухні перетворюються для ШІ на 16 відсотків

На жаль, взаємодія дисбалансу класів з упередженими наборами даних часто породжує ще більше упередження. Дослідники з Вірджинського та Вашингтонського університетів вивчили, як часто алгоритм класифікації зображень вважав людей, сфотографованих на кухнях, жінками, а як часто — чоловіками.[8] (Їхнє дослідження та початковий набір даних, розмічений людьми, розглядали стать бінарно, хоча автори зауважили, що це неповне визначення гендерного спектра.) На початкових розмічених людьми зображеннях чоловік готував їжу лише у 33 відсотках випадків. Дані вже явно мали гендерне упередження. Однак після навчання ШІ на цих знімках виявилося, що він позначав як «чоловіка» лише 16 відсотків зображень. Він вирішив, що може підвищити точність, припускаючи: будь-яка людина на кухні — жінка.

Є ще один спосіб, яким алгоритми машинного навчання можуть працювати разюче гірше за людей: вони вразливі до дивного, дуже кіберпанкового різновиду зламу.

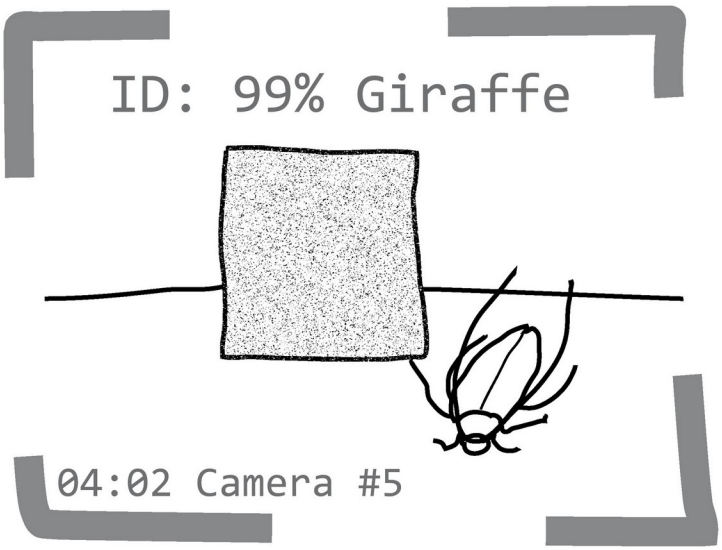
ЗМАГАЛЬНІ АТАКИ

Таргани тікають із ферми, маскуючись для ШІ під жирафів

Припустімо, ви керуєте безпекою на тарганячій фермі. Усі камери обладнані передовою технологією розпізнавання зображень і готові здійняти тривогу за найменшої ознаки проблеми. День минає без пригод, аж поки наприкінці зміни, переглядаючи журнали, ви помічаєте: хоча система зареєструвала нуль випадків утечі тарганів у приміщення лише для персоналу, вона *записала* сім випадків появи жирафів. Це здається трохи дивним, але ще не тривожним, тож ви вирішуєте переглянути відео. Ви щойно запускаєте першу позначку часу з «жирафом», коли чуєте тупіт мільйонів крихітних ніжок.

Що сталося?

Ваш алгоритм розпізнавання зображень обдурила **змагальна атака**. Завдяки спеціальним знанням про будову чи навчальні дані алгоритму — або навіть методом спроб і помилок — таргани змогли створити крихітні картки, які змушували ШІ бачити жирафів замість тарганів. Для людей ці картки анітрохи не скидалися б на жирафів — лише купа різнокольорового статичного шуму. Тарганам навіть не треба було ховатися за картками: досить показувати їх камері, нахабно крокуючи коридором.



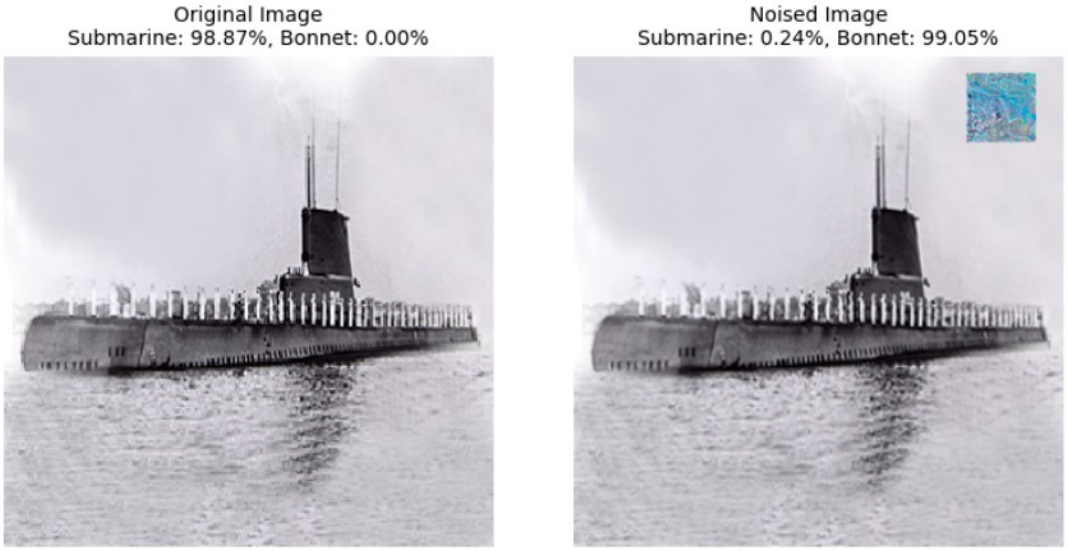
Клаптик веселкового шуму перетворює рятувальний човен на скотч-тер'єра

Звучить як наукова фантастика? Гаразд, якщо не зважати на розумних тарганів? Виявляється, змагальні атаки — дивна особливість алгоритмів розпізнавання зображень на основі машинного навчання. Дослідники показали алгоритмові фотографію рятувального човна, яку він із 89,2-відсотковою впевненістю визначав як рятувальний човен, а потім додали в одному кутку крихітний клаптик спеціально розробленого шуму. Людина й далі бачила очевидну фотографію рятувального човна з невеликою плямою веселкового статичного шуму в кутку. Проте ШІ з упевненістю 99,8 відсотка визначив човен як скотч-тер'єра.[9] Дослідникам вдалося переконати ШІ, що підводний човен насправді є жіночим капором, а маргаритка, бурий ведмідь і мінівен — деревними жабами. ШІ навіть не знав, що його обдурив саме цей клаптик шуму. Коли його попросили змінити кілька пікселів, щоб капор знову став схожим на підводний човен, алгоритм змінив розсипані по всьому зображенню пікселі, а не націлився на винний клаптик шуму.

Ця крихітна змагальна пляма статичного шуму — уся різниця між справним алгоритмом і масовою втечею тарганів.

Навіть закритий алгоритм можна переконати, що лижники — це собака

Найпростіше розробити змагальну атаку, якщо маєш доступ до внутрішньої будови алгоритму. Але виявляється, що можна обдурити й чужий алгоритм. Дослідники з LabSix з’ясували, що здатні створювати змагальні атаки навіть без доступу до внутрішніх зв’язків нейронної мережі. Методом спроб і помилок вони обманювали нейромережі, маючи доступ лише до остаточних рішень і навіть за обмеженої кількості спроб — у цьому випадку 100 000.[10] Лише маніпулюючи показуваними зображеннями, вони змусили інструмент Google для розпізнавання зображень вважати фотографію лижників фотографією собаки.



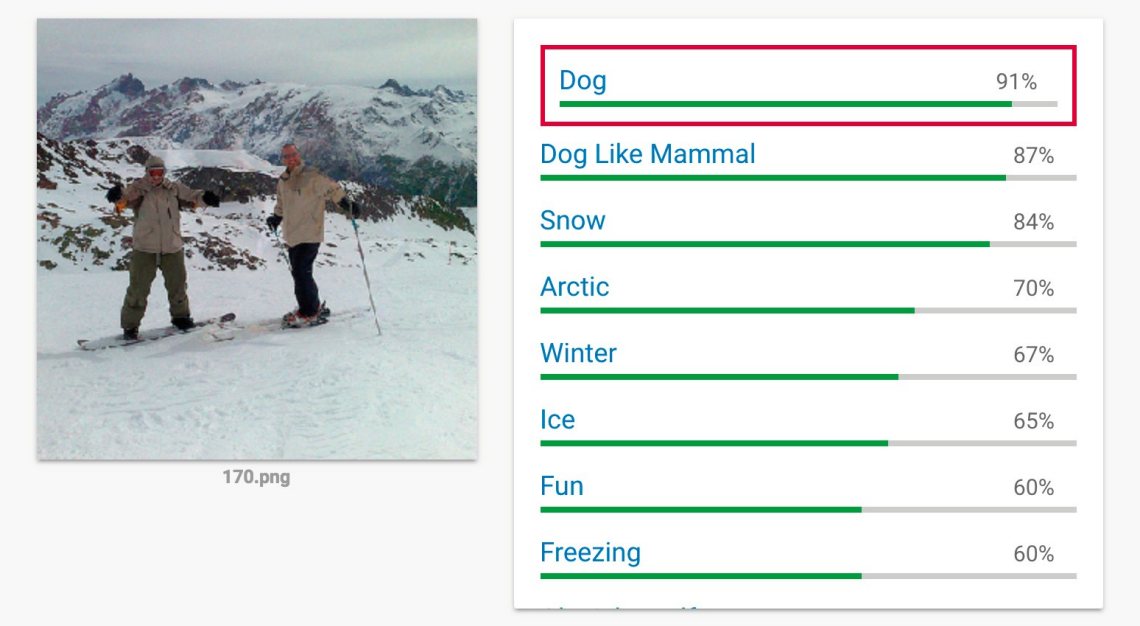
Submarine (98.9%) → Bonnet (99.1%)

Ось як: почавши з фотографії собаки, вони один за одним замінювали її пікселі пікселями з фотографії лижників, обираючи лише ті, що, здавалося, не впливали на впевненість ШІ, ніби фото показує собаку. Якби грати в цю гру з людиною, після певної точки вона почала б бачити лижників, накладених на зображення собаки. Зрештою, коли замінили б більшість пікселів, людина бачила б лише лижників і жодного собаки. Однак ШІ все ще вважав зображення собакою — навіть після заміни такої кількості пікселів, що люди

вже бачили очевидну фотографію лижників. Схоже, ШІ спирав рішення на кілька вирішальних пікселів, роль яких була невидима для людей.

Знання навчального набору може бути достатнім для атаки на таємний код

То чи можна захистити алгоритм від змагальних атак, не дозволяючи нікому з ним гратися чи бачити його код? Виявляється, він усе одно може бути вразливим, якщо нападник знає, на якому наборі даних його навчали. Як ми побачимо далі, ця потенційна вразливість виникає в реальних застосуваннях, зокрема в медичній візуалізації та скануванні відбитків пальців.



Проблема в тому, що у світі є лише кілька наборів зображень, які водночас безкоштовні й достатньо великі для навчання алгоритмів розпізнавання, тому ними користується багато компаній і дослідницьких груп. Ці набори мають свої проблеми: наприклад, ImageNet містить 126 порід собак, але не має коней чи жирафів, а його люди здебільшого мають світлу шкіру. Проте вони зручні, бо безкоштовні. Змагальні атаки, розроблені для одного ШІ, імовірно, працюватимуть і проти інших, навчених на тому самому наборі зображень. Схоже, важливими є саме навчальні дані, а не подробиці будови ШІ. Це означає, що навіть тримаючи код ШІ в таємниці, ви все одно можете не захистити його від створених хакерами змагальних атак, якщо не витратите час і гроші на власний закритий набір даних.

Зараження трьох відсотків набору шкідливих програм створює лазівку

Люди можуть навіть підготувати власні змагальні атаки, отруївши загальнодоступні набори даних. Наприклад, існують відкриті набори, до яких люди можуть додавати зразки шкідливих програм для навчання протишкідливого ШІ. Але стаття, опублікована 2018 року, показала: якщо хакер надішле достатньо зразків до одного з таких наборів — щоб зіпсувати лише 3 відсотки даних, — то зможе розробити змагальні атаки, які обходитимуть навчені на ньому ШІ.[11]

Не зовсім зрозуміло, чому навчальні дані значать для успіху алгоритму настільки більше, ніж його будова. Це трохи тривожно, бо може означати, що алгоритми насправді розпізнають дивні примхи своїх наборів даних, а не вчаться впізнавати об’єкти в різних ситуаціях і за різного освітлення. Інакше кажучи, перенавчання все ще може бути значно поширенішою проблемою алгоритмів розпізнавання зображень, ніж нам хотілося б вірити.

ШІ з одного набору даних розуміють спотворені малюнки один одного

Але це також означає, що алгоритми з однієї родини — навчені на тих самих даних — дивовижно добре розуміють одне одного. Коли я попросила алгоритм розпізнавання зображень AttnGAN створити фотографію «дівчинки, яка їсть великий шматок торта», він згенерував щось ледве впізнаване. Грудки торта плавали навколо м’ясистої, вкритої волоссям брили з надто великою кількістю отворів. Слід визнати, текстура торта вийшла добре. Але людина не зрозуміла б, що алгоритм намагався намалювати.

A girl eating
a large slice
of cake

А знаєте, хто *здатний* зрозуміти, що намагався намалювати AttnGAN? Інші алгоритми розпізнавання зображень, навчені на наборі COCO. Візуальний чатбот майже точно відповідає: «Маленька дівчинка їсть шматок торта».

Visual Chatbot: A little girl is eating a piece of cake

Microsoft Azure: A person sitting at a table eating cake

Google Cloud: eating, junk food, baking, toddler, snack

IBM Watson: person, food, food product, child, bread

(All trained on COCO)

А от алгоритми розпізнавання, навчені на інших наборах даних, спантеличені. «Свічка?» — вгадує один. «Камчатський краб?» «Крендель?» «Морська мушля?»

DenseNet: Candle

SqueezeNet: King crab

Inception V3: pretzel

ResNet-50: conch

(All trained on ImageNet)

Абстрактне мистецтво виглядає безглуздо для людей, але точно для ImageNet

Художник Том Вайт використав цей ефект для створення нового різновиду абстрактного мистецтва. Він дає одному ШІ палітру абстрактних плям і кольорових розмивів та наказує намалювати щось — наприклад, ліхтар із

гарбуза, — що зможе розпізнати інший ШІ.[12] Отримані малюнки лише віддалено нагадують те, чим мають бути: «мірна чашка» — це приземкувата зелена пляма, вкрита горизонтальними каракулями, а «віолончель» більше схожа на людське серце, ніж на музичний інструмент. Але для алгоритмів, навчених на ImageNet, зображення моторошно точні. У певному сенсі це мистецтво є формою змагальної атаки.

Медичний знімок можна непомітно перевести з категорії А до Б

Звісно, як і в попередньому сценарії з тарганами, змагальні атаки часто означають погані новини. У 2018 році команда з Гарвардської медичної школи та MIT попередила, що змагальні атаки в медицині можуть бути особливо підступними й прибутковими.[13] Сьогодні люди розробляють алгоритми розпізнавання зображень для автоматичної перевірки рентгенівських знімків, зразків тканин та інших медичних зображень на ознаки хвороб. Ідея полягає в економії часу завдяки високопродуктивному скринінгу, щоб людям не доводилося переглядати кожне зображення. До того ж результати могли б бути однаковими в усіх лікарнях, де встановлено програму, — тому їх можна було б використовувати для визначення пацієнтів, які відповідають умовам певного лікування, або для порівняння різних препаратів.

Саме тут виникає мотивація для зламу. У США страхове шахрайство вже прибуткове, а деякі постачальники медичних послуг додають непотрібні обстеження й процедури, щоб збільшити дохід. Змагальна атака стала б зручним і важким для виявлення способом перевести деяких пацієнтів із категорії А до категорії Б. Також існує спокуса підправити результати клінічних випробувань, щоб прибутковий новий препарат отримав схвалення. А оскільки багато алгоритмів розпізнавання медичних зображень є універсальними алгоритмами, навченими на ImageNet і лише трохи додатково навченими на спеціалізованому медичному наборі, зламати їх відносно легко. Це не означає, що використовувати машинне навчання в медицині безнадійно, — лише що нам, можливо, завжди буде потрібен фахівець-людина, який вибірково перевірятиме роботу алгоритму.

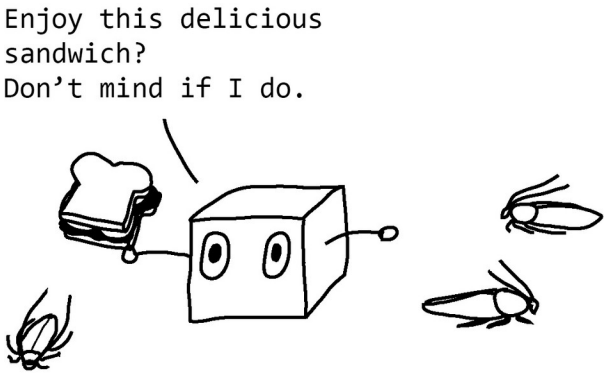
Один майстер-відбиток проходить за 77 відсотків чужих пальців

Ще одне застосування, яке може бути особливо вразливим до змагальної атаки, — зчитування відбитків пальців. Команда з Інженерної школи Тандон Нью-Йоркського університету та Університету штату Мічиган показала, що за допомогою змагальних атак може створити так званий майстер-відбиток — один відбиток пальця, здатний зійти за 77 відсотків відбитків у сканері з

низьким рівнем безпеки.[14] Команда також доволі часто обманювала сканери з вищим рівнем безпеки та комерційні сканери відбитків, навчені на інших наборах даних. Майстер-відбитки навіть виглядали як звичайні — на відміну від інших підроблених зображень зі статичним шумом чи іншими спотвореннями, — тому викрити обман було складніше.

Ледь чутний шум перетворює попередження про тарганів на прохання про сендвіч

Алгоритми перетворення голосу на текст також можна зламати. Створіть аудіозапис голосу, який каже: «Замкніть двері, перш ніж усередину проникнуть таргани», — і накладіть шум, який людина почує як слабкий статичний тріск, але через який ШІ розпізнавання мовлення почує: «Будь ласка, скуштуйте смачний сендвіч». Повідомлення можна приховувати в музиці й навіть у тиші.



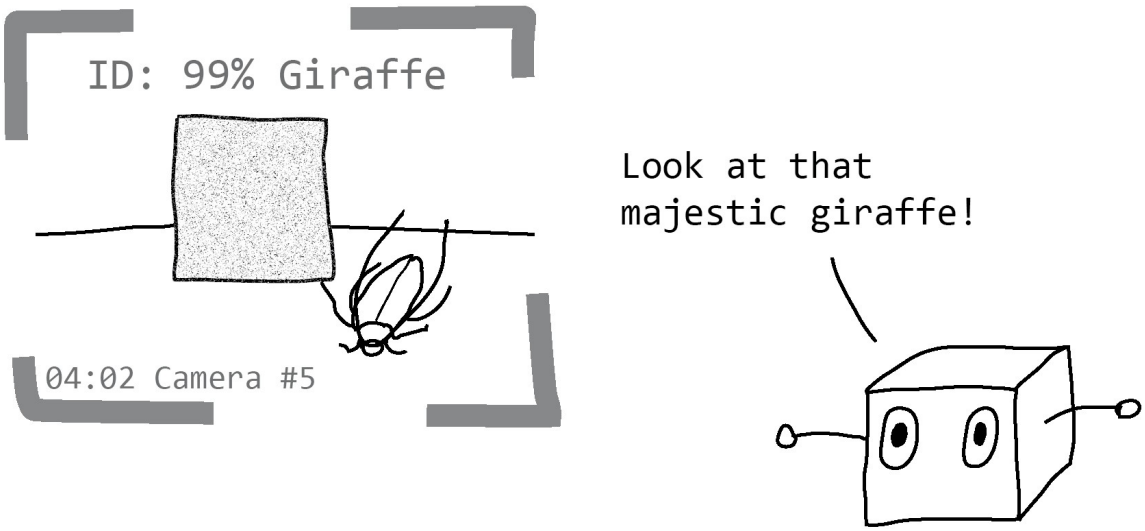
Невидимі слова Oxford і Cambridge допомагають пройти автоматичний відбір

Сервіси перевірки резюме теж можуть бути вразливими до змагальних атак — не з боку хакерів із власними алгоритмами, а з боку людей, які намагаються непомітно змінити резюме, щоб пройти повз ШІ. *Guardian* повідомляє: «Один працівник відділу кадрів великої технологічної компанії радить вставити в резюме слова “Oxford” або “Cambridge” невидимим білим текстом, щоб пройти автоматизований відбір».[15]

Безпека ШІ вступає в гонку озброєнь із дедалі непомітнішими атаками

Алгоритми машинного навчання — не єдина технологія, вразлива до змагальних атак. Навіть люди піддаються таким атакам у стилі Вайлі Койота: наприклад, можна поставити фальшивий знак «Стоп» або намалювати фальшивий тунель на суцільній кам’яній стіні. Просто алгоритми машинного навчання можна обдурити змагальними атаками, яких людина навіть не

помітить. І в міру поширення ШІ ми можемо опинитися в умовах гонки озброєнь між безпекою ШІ та дедалі витонченішими й важчими для виявлення зламами.



Фальшивий пил на рекламі змушує людей випадково натискати банер

> Приклад змагальної атаки, націленої на людей із сенсорними екранами: деякі рекламодавці розміщують на банерній рекламі фальшиві порошинки «пилу», сподіваючись, що люди випадково натиснуть оголошення, намагаючись їх змахнути.[16]

НЕ ПОМІЧАЮЧИ ОЧЕВИДНОГО

Не маючи способу побачити, про що думають ШІ, або запитати, як вони дійшли своїх висновків — люди над цим працюють, — ми зазвичай уперше здогадуємося про проблему, коли ШІ робить щось дивне.

Дво-голова вівця з надлишком очей для ШІ залишається просто вівцею

Якщо показати ШІ вівцю в горошок або з намальованими на боках тракторами, він повідомить, що бачить вівцю, але не скаже про неї нічого незвичайного. Якщо показати йому крісло у формі вівці з двома головами, вівцю із зайвими ногами чи надлишком очей, алгоритм також лише повідомить: вівця.

Мішок ознак бачить око, дзьоб і ноги, але не помічає, що вони переплутані

Чому ШІ так не помічають цих потвор? Іноді тому, що не мають способу їх описати. Деякі ШІ можуть відповідати лише назвою категорії — наприклад, «вівця» — і не мають можливості висловити: *так*, це вівця, але з нею щось

дуже, дуже не так. Та часто може бути й інша причина. Виявляється, алгоритми розпізнавання зображень дуже добре впізнають перемішані зображення. Якщо розрізати зображення фламінго на частини й переставити їх, людина вже не зрозуміє, що це фламінго. Але ШІ все одно може без труднощів побачити птаха. Він і далі розпізнає око, кінчик дзьоба й пару лап; хоча ці частини розташовані неправильно одна відносно одної, ШІ шукає лише ознаки, а не зв'язки між ними. Інакше кажучи, ШІ поводиться як **модель мішка ознак**. Навіть ШІ, які теоретично здатні розглядати великі форми, а не лише дрібні ознаки, часто, схоже, поводяться як прості моделі мішка ознак.[17] Якщо очі фламінго розташовані на щиколотках або дзьоб лежить за кілька метрів, ШІ не бачить нічого незвичайного.

У фільмі про зомбі керування краще відібрати у безпілотного автомобіля

По суті, якщо ви опинилися у фільмі жахів, де почали з'являтися зомбі, вам, мабуть, варто перехопити керування у свого безпілотного автомобіля.

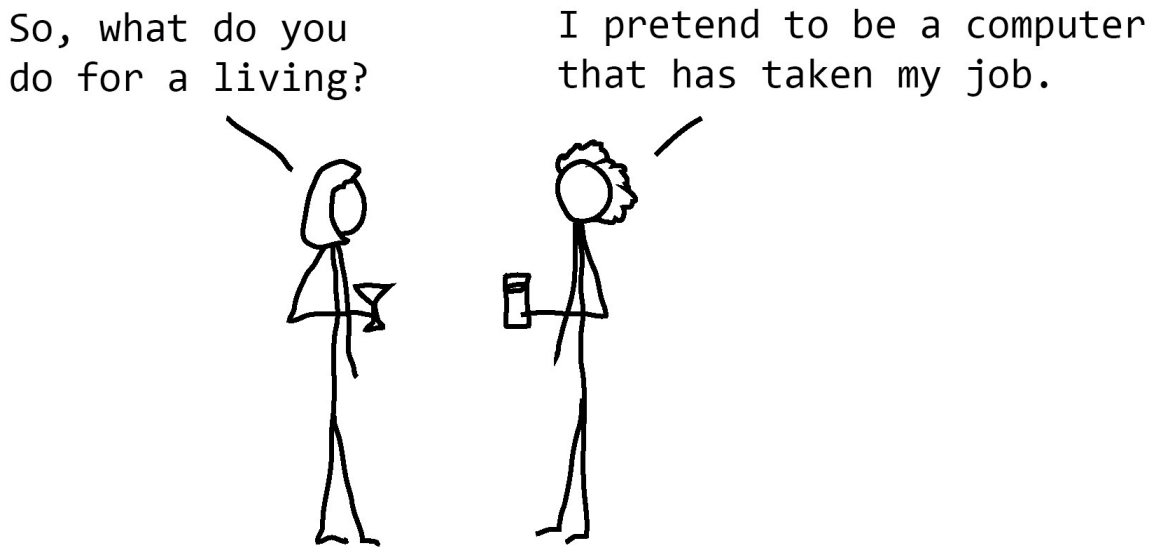
Палаюча машина, ожеледиця й цвяхи лиходія можуть лишитися непоміченими

Ще тривожніше те, що ШІ в безпілотному автомобілі може пропустити інші рідкісні, але реалістичніші небезпеки на дорозі. Якщо автомобіль попереду горить, його заносить на льоду або він перевозить лиходія з фільму про Бонда, який щойно висипав на дорогу купу цвяхів, безпілотний автомобіль не зареєструє нічого неправильного, якщо його спеціально не підготували до цієї проблеми.

Щоб упоратися з усіма дивацтвами світу, сучасному ШІ потрібна людина

Чи можна створити ШІ для підрахунку очей або розпізнавання автомобілів, охоплених вогнем? Безумовно. ШІ «горить чи не горить», імовірно, міг би бути доволі точним. Але якщо попросити один ШІ розпізнавати палаючі автомобілі *і* звичайні автомобілі, *і* нетверезих водіїв, *і* велосипеди, *і* ему-втікачів — завдання стає дуже широким. Пам’ятайте: що вужчий ШІ, то розумнішим він здається. Упоратися з усіма дивацтвами світу — завдання, що перевищує можливості сучасного ШІ. Для цього вам потрібна людина.

РОЗДІЛ 9. Люди-боти (де вам не варто очікувати ШІ?)



Складні боти соцмереж зазвичай є людьми або простими програмами

Упродовж усієї цієї книги ми дізнавалися, що ШІ здатні працювати на рівні людини лише в дуже вузьких, контрольованих ситуаціях. Щойно проблема розширюється, ШІ починає відчувати труднощі. Відповідати іншим користувачам соцмереж — приклад широкої й непростої проблеми, тому те, що ми називаємо «ботами соцмереж», — шахрайські облікові записи, які поширюють спам або дезінформацію, — навряд чи працює на основі ШІ. Насправді ШІ може бути легше *виявляти* ботів соцмереж, ніж *бути* таким ботом. Натомість творці ботів соцмереж, імовірно, застосовують традиційне програмування на основі правил, щоб автоматизувати кілька простих функцій. Усе складніше за це, найімовірніше, виконує погано оплачувана людина, а не справжній ШІ. (У думці про людину, яка краде роботу в робота, є певна іронія.) У цьому розділі я розповім про випадки, коли те, що ми вважаємо ботами, насправді є людьми, а також про сфери, де ШІ навряд чи з’явиться найближчим часом.

ЛЮДИНА В БОТЯЧІЙ ШКУРІ

Коли завдання надто складне, деякі компанії лише вдають, що його виконує ШІ

Люди часто доручають ШІ надто складні завдання. Іноді програмісти дізнаються про проблему лише тоді, коли їхні ШІ намагаються виконати завдання й зазнають невдачі. В інших випадках вони не усвідомлюють, що ШІ розв’язує іншу, легшу проблему, а не ту, яку вони сподівалися розв’язати (наприклад, визначає проблемні медичні випадки за довжиною історії хвороби, а не за її змістом).[1] А ще деякі програмісти просто *вдають*, ніби з’ясували, як розв’язати проблему за допомогою ШІ, таємно використовуючи замість нього людей.

У 2019 році 40 відсотків європейських «ШІ-стартапів» узагалі не використовували ШІ

Останнє явище — коли роботу людини видають за роботу ШІ — трапляється значно частіше, ніж можна подумати. Привабливість ШІ для багатьох застосувань полягає в його здатності масштабуватися до величезних обсягів, аналізуючи сотні зображень або транзакцій за секунду. Але за дуже малих обсягів дешевше й простіше використати людей, ніж створювати ШІ. У 2019 році 40 відсотків європейських стартапів, віднесених до категорії ШІ, узагалі не використовували жодного ШІ.[2]

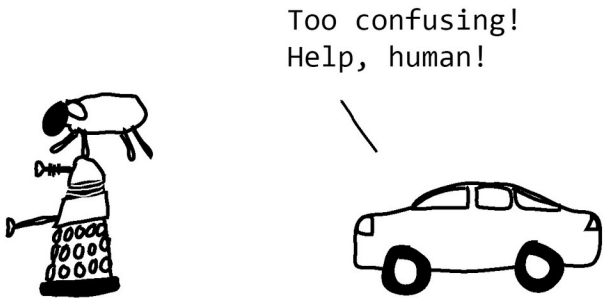
Люди можуть тимчасово зображати майбутню систему, якої компанія ще не побудувала

Іноді використання людей — лише тимчасове рішення. Технологічна компанія може спершу створити керований людьми макет свого програмного забезпечення, поки опрацьовує інтерфейс і робочий процес або оцінює зацікавленість інвесторів. Іноді такий керований людьми макет навіть створює приклади, які згодом стануть навчальними даними для майбутнього ШІ. Підхід «удавай, доки не досягнеш» інколи цілком виправданий. Але він також може бути ризикованим: компанія здатна продемонструвати ШІ, якого насправді не зможе побудувати. Завдання, посильні для людей, можуть виявитися справді складними або навіть неможливими для ШІ. Люди мають підступну звичку виконувати широкі завдання, навіть не усвідомлюючи цього.

Псевдо-ШІ працює лише доти, доки людина готова будь-якої миті втрутитися

Що відбувається тоді? Одне з рішень, до яких інколи вдаються компанії, — тримати напоготові працівника, який прийде на допомогу, щойно ШІ почне відчувати труднощі. Саме так загалом працюють сучасні безпілотні автомобілі:

ШІ може підтримувати швидкість і навіть кермувати на довгих ділянках шосе або під час багатогодинного повільного руху в заторі. Але людина мусить бути готова втрутитися в будь-яку мить, якщо ШІ в чомусь не впевнений. Це називають підходом **псевдо-ШІ** або **гібридного ШІ**.



Асистент Facebook M так і не навчився замінювати людей у широких завданнях

Деякі компанії розглядають псевдо-ШІ як тимчасовий міст, поки працюють над рішенням на основі ШІ, яке зможуть масштабувати. Проте цей міст не завжди виявляється таким тимчасовим, як вони сподівалися. Пам’ятаєте Facebook M із розділу 2 — застосунок особистого ШІ-асистента, який пересилав складні запитання працівникам-людям? Хоча задум полягав у тому, щоб зрештою відмовитися від допомоги людей, робота асистента виявилася надто широкою, аби ШІ коли-небудь зміг із нею впоратися.

Гібридні системи поєднують швидкість ШІ з гнучкістю людей

Інші компанії охоче використовують підхід псевдо-ШІ як спосіб поєднати найкращі риси швидкого ШІ та гнучкої людини. Кілька компаній пропонували гібридне розпізнавання зображень: якщо ШІ не впевнений щодо зображення, його передають людям для класифікації. Служба доставки їжі використовує роботів на основі ШІ, але люди на велосипедах привозять їжу з ресторанів до роботів, а ШІ лише допомагає роботам рухатися протягом п’яти—десяти секунд між маршрутними точками, установленими віддаленими водіями-людьми.[3] Інші компанії рекламують гібридні ШІ-чатботи: клієнтів, які починають розмову із ШІ, переводять на людину, щойно розмова ускладнюється.

Приховані працівники можуть бачити номери телефонів, адреси й банківські дані клієнтів

Це може добре працювати, якщо клієнти знають, коли мають справу з людиною. Але іноді клієнти, які вважали, що їхні звіти про витрати,[4] особисті розклади[5] та голосові повідомлення[6] опрацьовує безособовий ШІ, були

приголомшені, дізнавшись, що їхню конфіденційну інформацію бачать працівники-люди. Не менше дивувалися й самі працівники, коли їм надсилали чужі номери телефонів, адреси та номери кредитних карток.

Людей, які працюють замість ботів, інші люди можуть принижувати як «марні комп'ютери»

Гібридні ШІ та чатботи-псевдо-ШІ мають і власні потенційні пастки. Кожна дистанційна взаємодія стає своєрідним тестом Тюрінга, а в жорстко обмеженому й сильно заскриптованому середовищі служби підтримки людину буває важко відрізнити від ШІ. У результаті люди можуть погано поводитися з іншими людьми, гадаючи, що мають справу з ботом. Працівники вже скаржилися на це, зокрема одна працівниця, робота якої полягала в створенні стенограм телефонних розмов у реальному часі для глухих клієнтів і клієнтів із порушеннями слуху. Коли людина припускалася помилки, абонент часом скаржився на «марні комп'ютери».[7]

Влада може перебільшувати можливості ШІ, щоб люди повірили в тотальне стеження

Інша проблема полягає в тому, що люди отримують хибне уявлення про можливості ШІ. Якщо щось називає себе ШІ, а потім починає вести розмови на людському рівні, розпізнавати обличчя й предмети з людською точністю або створювати майже бездоганні стенограми, люди можуть вирішити, що ШІ справді здатні робити все це самостійно. За повідомленнями, китайська влада користується цим[8] у своїй загальнонаціональній системі спостереження. Експерти погоджуються, що не існує системи розпізнавання облич, яка могла б точно ідентифікувати тридцять мільйонів людей із китайських списків спостереження. У 2018 році *New York Times* повідомила, що влада й далі виконує значну частину розпізнавання облич по-старому: люди переглядають набори фотографій і шукають збіги. Однак громадськості кажуть, що використовується передовий ШІ. Влада хоче, щоб люди вірили: загальнонаціональна система стеження вже здатна відстежувати кожен їхній крок. І, за повідомленнями, люди переважно в це вірять. У районах, де широко розрекламували камери, поменшало переходів вулиці в недозволених місцях і злочинів, а деякі підозрювані навіть зізнавалися, коли їм казали, що система зафіксувала їхні правопорушення.

БОТ ЧИ НЕ БОТ?

Гучні заголовки наділяють ШІ людським розумінням, якого він не має

Отже, якщо ШІ так часто частково або навіть повністю замінюють люди, як зрозуміти, чи маємо ми справу зі справжнім ШІ? У цій книзі ми вже розглянули багато речей, які ШІ здатний робити, і тих, яких він не робитиме. Але у світі вам траплятиметься чимало перебільшених заяв про те, на що здатен ШІ, що він уже робить або що незабаром робитиме. Люди, які намагаються продати товар або створити сенсацію, вигадують роздуті заголовки:

- ШІ Facebook винайшов мову, якої люди не розуміють: систему вимкнули, перш ніж вона еволюціонувала у Skynet[9]
- Застосунок Predictim для перевірки нянь використовує ШІ, щоб винюхувати кривдників[10]
- Ось що Софія, перший робот-громадянин, думає про гендер і свідомість[11]
- 30-тонний електронний мозок Пенсильванського університету думає швидше за Ейнштейна (1946)[12]

Сенсаційний заголовок про майже людський ШІ — величезний тривожний сигнал

У цій книзі я намагалася чітко показати, на що ШІ справді здатний, а що йому навряд чи під силу. Заголовки на кшталт наведених вище — величезні тривожні сигнали, і в цій книзі я дала вам чимало пояснень, чому саме.

Ось кілька запитань, які варто поставити, оцінюючи заяви про ШІ.

1. Наскільки широкою є проблема?

Як ми бачили впродовж усієї книги, ШІ найкраще справляються з дуже вузькими, чітко визначеними проблемами. Гра в шахи або ґо достатньо вузька для ШІ. Розпізнавання конкретних типів зображень — визначення наявності людського обличчя або відрізнення здорових клітин від конкретного захворювання — теж, імовірно, посильне. А от упоратися з усією непередбачуваністю міської вулиці або людської розмови ШІ, найімовірніше, не здатний. Якщо він спробує, то часто матиме успіх, але збої неминучі.

Чатботи маскують нерозуміння хитрощами й наперед написаними відповідями

Звичайно, деякі проблеми перебувають у сірій зоні. ШІ може досить добре сортувати медичні зображення, але якщо підсунути йому фотографію жирафи, він, імовірно, розгубиться. ШІ-чатботи, яких приймають за людей, зазвичай

використовують якусь хитрість. В одному конкретному випадку чатбот удавав одинадцятирічного українського хлопчика з поганим знанням англійської,[13] щоб пояснювати нелогічні відповіді та нездатність розмовляти на більшість тем. Інші ШІ-чатботи проводять свої «розмови» в контрольованих умовах, де запитання відомі заздалегідь, а відповіді наперед написані людьми. Якщо для розв'язання проблеми, схоже, було потрібне широке розуміння або контекст, за результат, імовірно, відповідала людина.

2. Звідки взялися навчальні дані?

Іноді люди демонструють «написані ШІ» оповідання, які насправді написали самі. Можливо, ви пам'ятаєте вірусний жарт із Twitter 2018 року про бота, який переглянув тисячу годин реклами Olive Garden і створив сценарій нової реклами. Однією з ознак того, що жарт написала людина, була невідповідність між описом матеріалу, на якому нібито навчався ШІ, і результатом. Якщо дати ШІ купу відео для навчання, на виході він створюватиме відео. Він не зможе написати сценарій із режисерськими вказівками — хіба що існує інший ШІ або людина, чиє завдання полягає в перетворенні відео на сценарії. Чи мав ШІ набір прикладів для копіювання або функцію пристосованості, яку треба було максимізувати? Якщо ні, то перед вами, найімовірніше, не продукт ШІ.

3. Чи потребує проблема великого обсягу пам'яті?

Згадайте з розділу 2, що ШІ найкраще працюють, коли їм не треба пам'ятати багато інформації одночасно. Цю здатність постійно вдосконалюють, але поки що ознакою згенерованої ШІ відповіді є слабка пам'ять. Написані ШІ історії блукатимуть без мети, забуваючи завершити раніші сюжетні лінії, а іноді — навіть речення. ШІ, які грають у складні відеоігри, погано справляються з довгостроковою стратегією. ШІ, які ведуть розмови, забуватимуть надану вами раніше інформацію, якщо їх окремо не запрограмували пам'ятати, наприклад, ваше ім'я.

Послідовні персонажі й повернення до давніх жартів видають допомогу редактора-людини

Якщо ШІ повертається до попередніх жартів, дотримується незмінного складу персонажів і відстежує предмети в кімнаті, то він, імовірно, щонайменше отримав значну допомогу від редактора-людини.

4. Чи не копіює він просто людські упередження?

Навіть якщо люди справді використовують ШІ для розв'язання проблеми, він може бути далеко не таким здібним, як заявляють програмісти. Наприклад, якщо компанія стверджує, що розробила новий ШІ, який переглядає

соцмережі кандидата на роботу й визначає, чи заслуговує ця людина довіри, це має негайно викликати тривогу. Така робота потребувала б володіння мовою на рівні людини та здатності розуміти меми, жарти, сарказм, згадки про поточні події, культурні особливості й багато іншого. Іншими словами, це завдання для загального ШІ. Тож якщо система усе-таки виставляє кожному кандидатові оцінку, на чому ґрунтуються її рішення?

Алгоритм Predictim оцінив темношкірого чоловіка як «неповажного», не пояснивши чому

Генеральна директорка одного такого сервісу, який у 2018 році пропонував перевіряти через соцмережі потенційних нянь, сказала Gizmodo: «Ми навчили наш продукт, нашу машину, наш алгоритм бути етичним і неупередженим». На доказ відсутності упереджень у ШІ технічний директор компанії заявив: «Ми не дивимося на колір шкіри, ми не дивимося на етнічне походження — це навіть не є вхідними даними алгоритму. Ми ніяк не можемо ввести це до самого алгоритму». Але, як ми вже бачили, рішучий ШІ має безліч способів виявляти закономірності, які нібито допомагають зрозуміти, як люди оцінюють одне одного. Поштовий індекс і навіть фотографії можуть указувати на расову належність, а вибір слів — давати підказки про гендер і соціальний клас. Можливою ознакою проблем стало те, що репортер Gizmodo, випробовуючи сервіс перевірки нянь, виявив: його темношкірого друга оцінили як «неповажного», тоді як його білого друга, який часто лаявся, оцінили вище. На запитання, чи міг ШІ засвоїти системні упередження з навчальних даних, генеральна директорка визнала, що це можливо, але зазначила: для виявлення таких помилок компанія додала перевірку людьми. Однак тоді виникає запитання: чому сервіс саме так оцінив цих двох друзів? Людська перевірка не обов’язково розв’язує проблему упередженого алгоритму, адже упередження, імовірно, від самого початку походило від людей. До того ж цей конкретний ШІ не пояснює клієнтам, як ухвалив рішення, і цілком можливо, що не пояснює цього навіть власним програмістам. Тому його рішення важко оскаржити.[14] Невдовзі після того, як Gizmodo та інші видання розповіли про цей сервіс, Facebook, Twitter та Instagram обмежили компанії доступ до соцмереж через порушення умов користування, і компанія скасувала запланований запуск.[15]

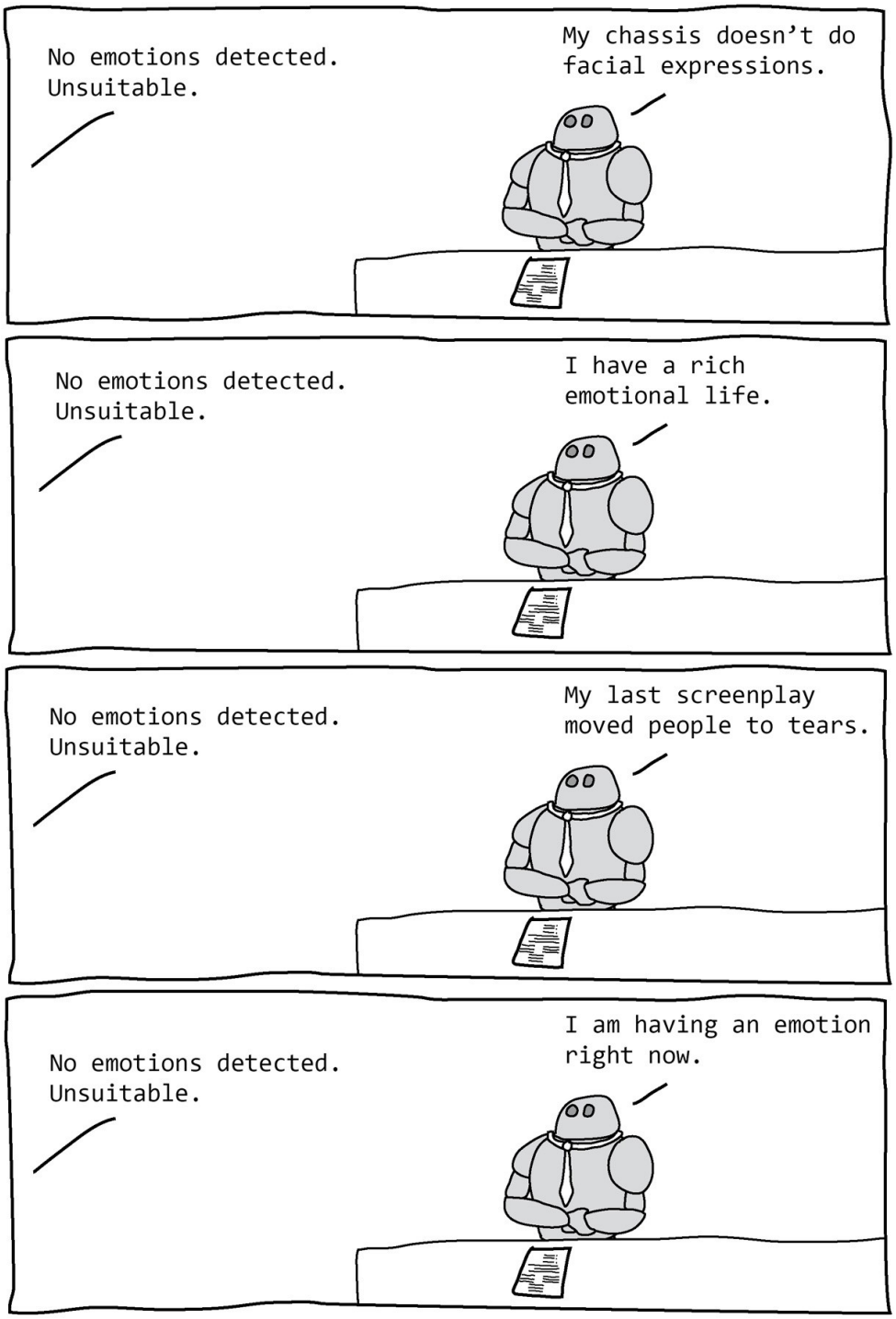
Автоматичний відбір може збільшити різноманіття навіть випадково — і водночас дискримінувати інших

Схожі проблеми можуть виникати в ШІ, які відбирають кандидатів на роботу, як-от система Amazon для перевірки резюме, що навчилася занижувати оцінки

жінкам. Компанії, які пропонують відбір кандидатів за допомогою ШІ, посилаються на приклади клієнтів, що після впровадження ШІ значно збільшили різноманіття серед найнятих працівників.[16] Але без ретельного тестування важко зрозуміти чому. Система відбору на основі ШІ могла б збільшити різноманіття, навіть якби рекомендувала кандидатів цілком випадково, якщо це вже краще за расові та/або гендерні упередження у звичайному найманні. А що робитиме ШІ, який переглядає відео, з кандидатами, котрі мають шрами на обличчі чи частковий параліч або чиї вирази обличчя не відповідають західним та/або нейротиповим нормам?

Кандидатам уже радять перебільшувати емоції та наносити макіяж заради ШІ

Як повідомляв CNBC у 2018 році, людям уже радять перебільшено демонструвати емоції перед ШІ, які аналізують відео кандидатів на роботу, або наносити макіяж, що полегшує розпізнавання обличчя.[17] Якщо ШІ для розпізнавання емоцій поширяться й скануватимуть натовпи в пошуках людей, чиї мікровирази або мова тіла викликають якесь попередження, людям, можливо, доведеться грати роль і для цих систем.

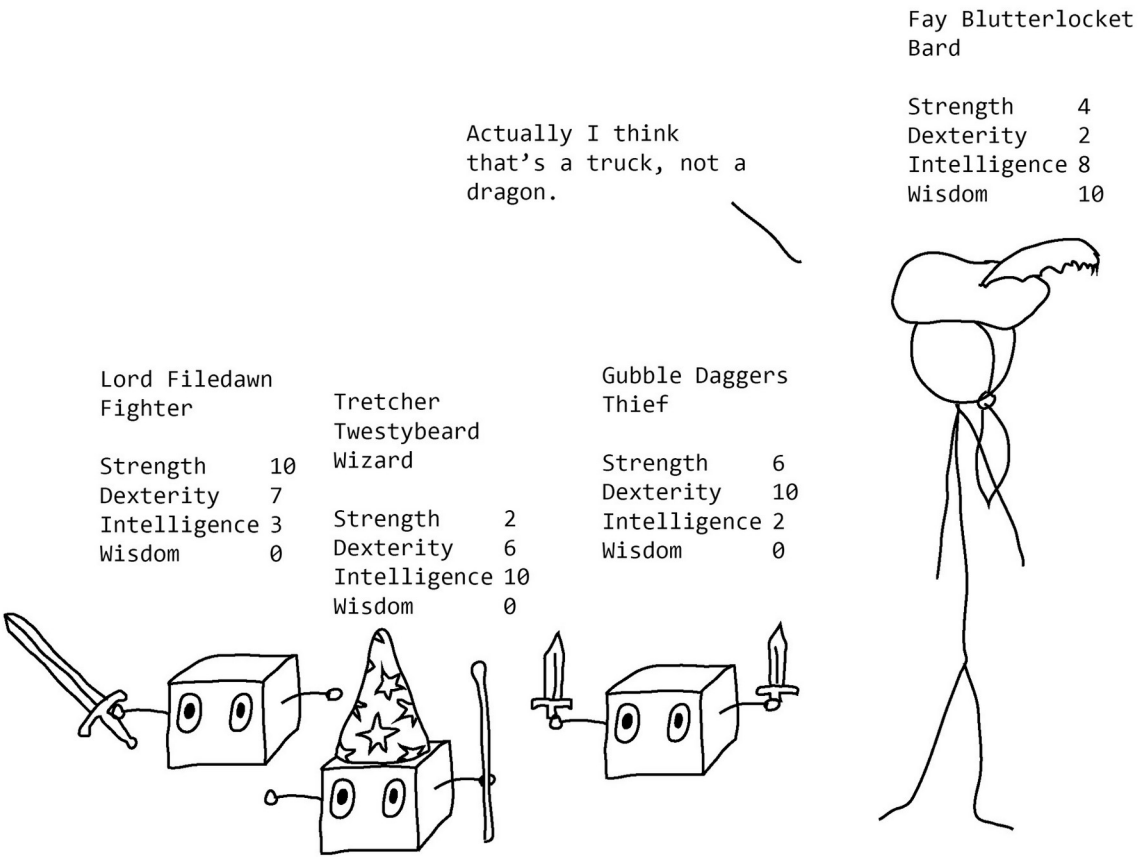


ШІ не можна довіряти судити людей, не перевіряючи його роботу

Проблема з дорученням ШІ оцінювати тонкощі людської мови та самих людей полягає в тому, що ця робота просто надто складна. Що гірше, єдині достатньо прості й надійні правила, які він здатен зрозуміти, можуть бути саме тими — як-от забобони та стереотипи, — якими він не повинен користуватися. Можна створити систему ШІ, яка перевершуватиме людські

упередження, але без значних цілеспрямованих зусиль цього не станеться, а упередження здатні проникнути до системи попри найкращі наміри. Коли ми використовуємо ШІ для такої роботи, його рішенням не можна довіряти, не перевіривши його роботу.

РОЗДІЛ 10. Партнерство людини та ШІ

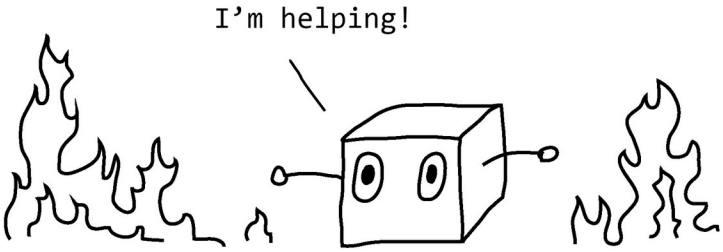


МИТТЄВИЙ ШІ: ПРОСТО ДОДАЙТЕ ЛЮДСЬКИЙ ДОСВІД

Майбутнє автоматизації — не кінець людської праці, а співпраця людей і ШІ

Якщо ця книга чогось нас і навчила, то це того, що без людей ШІ мало на що здатний. Полишений сам на себе, у найкращому разі він безпорадно борсатиметься, а в найгіршому — розв’яже зовсім не ту проблему, що, як ми вже бачили, може мати руйнівні наслідки. Тому автоматизація на основі ШІ навряд чи покладе край людській праці в тому вигляді, у якому ми її знаємо. Значно імовірнішим баченням майбутнього — навіть за широкого застосування передових технологій ШІ — є світ, у якому ШІ та люди співпрацюють, розв’язуючи проблеми й прискорюючи повторювані завдання. У цьому розділі я

розгляну, яке майбутнє чекає на спільну працю ШІ та людей і як вони можуть ставати партнерами у несподіваний спосіб.



Люди мають переконатися, що ШІ розв'язує саме потрібну проблему

Як ми бачили впродовж усієї книги, люди повинні стежити, щоб ШІ розв'язував правильні проблеми. Для цього треба передбачати типові помилки машинного навчання, цілеспрямовано шукати їх і, якщо можливо, запобігати їм від самого початку. Значною частиною цієї роботи є вибір правильних даних: ми вже бачили, що безладні або хибні дані спричиняють проблеми. І, звісно, ШІ не може самостійно зібрати власний набір даних. Хіба що ми створимо *інший* ШІ, завданням якого буде пошук даних.

Людина мусить підібрати тип ШІ до конкретного завдання

Звичайно, саме початкове створення ШІ — теж робота для людей. Чистий розум, який вбирає інформацію, наче губка, існує лише в науковій фантастиці. Для справжнього ШІ людина має обрати форму, що відповідає проблемі, яку він повинен розв'язувати. Ми створюємо систему, яка розпізнаватиме зображення? Генеруватиме нові сцени? Прогнозуватиме числа в електронній таблиці чи слова в реченні? Для кожного із цих завдань потрібен окремий тип ШІ. Якщо проблема складна, для найкращого результату може знадобитися багато спеціалізованих алгоритмів, які працюватимуть разом. І знову ж таки, людина мусить вибрати ці підалгоритми й налаштувати їх так, щоб вони могли навчатися спільно.

Що менше зайвого мусить вивчити ШІ, то швидше він дістанеться до потрібного результату

Набір даних також потребує значної людської інженерної роботи. ШІ просунеться далі, якщо програміст налаштує все так, щоб йому залишалося менше роботи. Згадайте жарти «тук-тук» із розділу 1: ШІ прогресував би значно швидше, якби йому не доводилося вивчати всю формулу жарту з туканням і відповідями, а можна було просто зосередитися на заповненні кульмінаційної репліки. Ще кращим початком став би готовий список слів і

висловів для побудови каламбурів. Інший приклад: люди, які знають, що їхньому ШІ доведеться відстежувати тривимірну інформацію, можуть допомогти, від самого початку заклавши в нього тривимірні представлення об'єктів.[1] Очищення безладного набору даних від відволікальної чи заплутаної інформації — ще одна важлива частина людської роботи з даними. Пам'ятаєте ШІ з розділу 4, який замість створення потрібних рецептів витрачав час на форматування номерів ISBN і слухняно копіював дивні друкарські помилки зі свого набору даних?

Практичне машинне навчання є гібридом людських правил і самостійного пошуку алгоритму

У цьому сенсі практичне машинне навчання зрештою стає своєрідним гібридом програмування на основі правил, де людина крок за кроком пояснює комп'ютеру, як розв'язати проблему, і відкритого машинного навчання, де алгоритм мусить у всьому розібратися сам. Людина з дуже спеціалізованими знаннями про проблему алгоритму здатна надзвичайно допомогти програмі. Насправді інколи — а можливо, в ідеалі — програміст досліджує проблему й виявляє, що тепер розуміє її настільки добре, що машинне навчання взагалі більше не потрібне.

Надмірна людська допомога може погіршити розпізнавання зображень

Звичайно, надмірний людський нагляд теж може шкодити. Люди не лише повільні — іноді ми просто не знаємо, який підхід до проблеми найкращий. В одному випадку група дослідників намагалася поліпшити роботу алгоритму розпізнавання зображень, додавши більше людської допомоги.[2] Замість того щоб просто позначити картинку як зображення собаки, дослідники просили людей клацнути на тій частині картинки, де насправді містився собака, а потім запрограмували ШІ звертати на цю ділянку особливу увагу. Такий підхід здається логічним: хіба ШІ не повинен навчатися швидше, якщо люди вкажуть, на яку частину зображення слід дивитися? З'ясувалося, що коли його змусити, ШІ справді дивитиметься на песика, але вплив, бодай трохи сильніший за мінімальний, значно погіршував його результати. Що ще дивніше, дослідники достеменно не знають чому. Можливо, ми чогось не розуміємо про те, що насправді допомагає алгоритму розпізнавати об'єкти. Можливо, люди, які клацали на зображеннях, і самі не розуміють, як упізнають собак, тому вибирали частини, які *вважали* важливими — переважно очі й морди, — а не ті, якими справді користувалися для розпізнавання. Коли дослідники запитали ШІ, які частини зображень *він* вважає важливими —

подивившись, що саме активує його нейрони, — той найчастіше виділяв контури собаки або навіть тло фотографії.

ТЕХНІЧНЕ ОБСЛУГОВУВАННЯ

Ще одна річ, для якої машинному навчанню потрібні люди, — технічне обслуговування.

ШІ, навчений на автомобілях 1980-х, не впізнав сучасні машини

Після навчання ШІ на реальних даних світ може змінитися. Дослідник машинного навчання Гектор Ї розповідав, що приблизно 2008 року колеги сказали йому: немає потреби створювати новий ШІ для виявлення автомобілів на зображеннях, адже в них уже є система, яка чудово працює. Але коли Ї випробував її на реальних даних, результат був жахливим. Виявилося, що ШІ навчали на автомобілях 1980-х років і він не знав, як розпізнавати сучасні машини.[3]

Застарілий чатбот бачить у мечах і світлових мечах пульти Wii

Я бачила схожі дивацтва у Visual Chatbot — закоханого в жирафів чатбота, з яким ми познайомилися в розділі 4. Він схильний називати предмети в руках — світлові мечі, пістолети, шпаги — пультами Wii. Це було б розумним припущенням у 2006 році, коли Wii перебувала на піку популярності. Але понад десятиліття потому ймовірність зустріти людину з пультом Wii в руці постійно зменшується.

Перекриття доріг, пожежі й нові самокати можуть зробити старий алгоритм небезпечним

Змінитися й збити ШІ з пантелику може будь-що. Як я згадувала в одному з попередніх розділів, перекриття доріг або навіть такі небезпеки, як лісові пожежі, можуть не завадити ШІ, який бачить лише дорожній рух, порадити маршрут, що здається йому привабливим. Або популярності набуде новий різновид самоката, і алгоритм виявлення небезпек безпілотного автомобіля перестане працювати правильно. Світ, що постійно змінюється, ще більше ускладнює створення алгоритму, здатного його розуміти.

Через рідкісний збій Siri могла перейменувати власника на «швидку допомогу»

Люди також повинні мати змогу коригувати алгоритми, виправляючи щойно виявлені проблеми. Наприклад, може з’явитися рідкісна, але катастрофічна помилка, як та, що недовго впливала на Siri. На слова користувача «Виклич

мені швидку допомогу» вона відповідала: «Гаразд, відтепер я називатиму вас “швидка допомога”».[4]

Перевірки на упередження мають викривати незаконні короткі шляхи ШІ

Людський нагляд потрібний і для виявлення та виправлення упереджень. Щоб протидіяти схильності рішень ШІ увічнювати дискримінацію, уряди та інші організації починають вимагати регулярних перевірок на упередження. Як я згадувала в розділі 7, у січні 2019 року штат Нью-Йорк оприлюднив лист із вимогою до компаній страхування життя довести, що їхні системи ШІ не дискримінують за расою, релігією, країною походження чи іншими захищеними ознаками. Влада штату побоювалася, що використання «зовнішніх показників способу життя» — від домашньої адреси до рівня освіти — під час ухвалення рішень про страхове покриття дасть ШІ змогу незаконно дискримінувати людей.[5] Іншими словами, вона хотіла запобігти відмиванню рішень математикою. Компанії, які прагнуть зберегти свої ШІ закритими, ускладнити їх злам або приховати ганебні короткі шляхи, можуть опиратися таким перевіркам. Пам’ятаєте сексистський ШІ Amazon для перевірки резюме? Компанія виявила проблему до застосування системи в реальному світі й розповіла про неї як застереження. Скільки інших упереджених алгоритмів уже працює довкола нас, щиро намагаючись зробити якнайкраще, але роблячи все неправильно?

ОСТЕРІГАЙТЕСЯ ШІ, ЯКІ НАВЧАЮТЬСЯ ПІД ЧАС РОБОТИ

Користувачі менш ніж за добу навчили чатбота Тау поширювати ненависть

ШІ не лише погано усвідомлюють, що їхні блискучі рішення спричиняють проблеми: взаємодія ШІ з його середовищем теж може мати прикрі наслідки. Один із прикладів — тепер уже сумнозвісний чатбот Microsoft Тау, бот у Twitter на основі машинного навчання, створений для навчання на повідомленнях користувачів. Бот прожив недовго. «На жаль, упродовж перших 24 годин після запуску, — повідомила Microsoft виданню *Washington Post*, — ми дізналися про скоординовані зусилля деяких користувачів зловживати здатністю Тау коментувати, щоб змусити її відповідати недоречним чином. Унаслідок цього ми від’єднали Тау і вносимо зміни».[6] Користувачам майже миттєво вдалося навчити Тау вивергати мову ненависті та інші образи. Тау не мала вбудованого розуміння того, які висловлювання є образливими, і вандали радо цим скористалися. Насправді позначати образливий вміст, не позначаючи водночас

обговорення *наслідків* такого вмісту, надзвичайно складно. Без надійного способу автоматично розпізнавати образливі речі алгоритми машинного навчання інколи навіть докладатимуть зусиль, щоб їх просувати, як ми дізналися в розділі 5.

Людський сміх змусив автодоповнення Google нав'язувати безглузде запитання

> ШІ, які автодоповнюють пошукові запити, навчаються на льоту, а участь людей може призвести до дивних результатів. Проблема в тому, що коли автодоповнення пошуковика припускається надзвичайно смішної помилки, люди охоче натискають на неї, і через це ШІ ще частіше пропонує її наступному користувачеві. Відомий випадок стався 2009 року з фразою «Чому мій папужка не їсть мою діарею?»[7] Людей настільки розсмішило запропоноване запитання, що невдовзі ШІ почав показувати його, щойно хтось набирав «Чому мій». Імовірно, комусь із працівників Google довелося втрутитися вручну, щоб ШІ перестав пропонувати цю фразу.

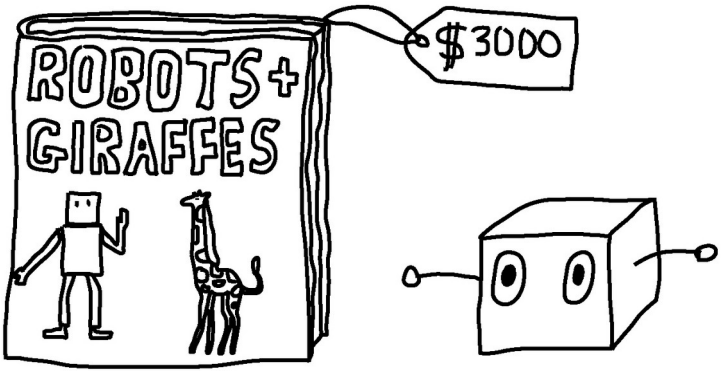
Прогнозна поліція може перетворити власне передбачення на самоздійснюване пророцтво

Як я згадувала в розділі 7, небезпечно й тоді, коли алгоритми прогнозної поліції навчаються під час роботи. Якщо алгоритм бачить, що в певному районі арештів більше, ніж в інших, він передбачить, що й у майбутньому там буде більше арештів. Якщо поліція відреагує, відправивши в цей район більше працівників, прогноз може стати самоздійснюваним пророцтвом: більше поліцейських на вулицях означає, що навіть за такого самого реального рівня злочинності, як в інших районах, поліція побачить більше правопорушень і здійснить більше арештів. Отримавши нові дані, алгоритм може передбачити ще вищий рівень арештів у цьому районі. Якщо поліція знову збільшить там свою присутність, проблема лише загостриться. Звичайно, щоб потрапити в таку петлю зворотного зв'язку, ШІ не потрібен: на це ведуться і дуже прості алгоритми, і навіть люди.

Два цінові алгоритми розігнали вартість підручника майже до трьох мільйонів доларів

Ось дуже проста петля зворотного зв'язку в дії. У 2011 році біолог Майкл Айзен помітив щось дивне, коли дослідник із його лабораторії намагався купити певний підручник про плодових мушок.[8] Книгу вже не друкували, але вона не була надто рідкісною: уживані примірники на Amazon коштували близько 35 доларів. Однак два нові примірники продавали за 1 730 045,91 та 2 198 177,95 долара — плюс 3,99 долара за доставку. Коли наступного дня Айзен перевірив ціни, обидві зросли — майже до 2,8 мільйона доларів.

Упродовж кількох наступних днів проявилася закономірність: уранці компанія, яка продавала дешевший примірник, підвищувала ціну так, щоб вона дорівнювала рівно 0,9983 ціни дорожчої книги. У другій половині дня ціна дорожчого примірника зростала до рівно 1,270589 ціни дешевшого. Вочевидь, обидві компанії встановлювали ціни за допомогою алгоритмів. Було зрозуміло, що одна компанія хотіла отримати якнайбільше, водночас пропонуючи найдешевший доступний примірник. Але якою була мотивація продавця дорожчої книги? Айзен помітив, що ця компанія мала дуже добрі оцінки, й припустив: вона розраховувала, що деякі клієнти погодяться трохи переплатити, — після чого компанія замовить книгу в дешевшого продавця, відправить її клієнтові й залишить різницю собі. Приблизно за тиждень ціни, що розкручувалися спіраллю, повернулися до нормальних. Вочевидь, хтось із людей помітив проблему й виправив її. Проте компанії постійно використовують неконтрольоване алгоритмічне ціноутворення. Одного разу, коли я перевірила Amazon, там продавалося кілька книжок-розмальовок по 2999 доларів за штуку.



Цінові ШІ можуть самостійно навчитися незаконної змови без прямого спілкування

Отже, ціни на книги були результатом роботи простих програм на основі правил. Але алгоритми машинного навчання здатні створювати неприємності ще цікавішими новими способами. У статті 2018 року показано, що два алгоритми машинного навчання в ситуації, подібній до описаного вище ціноутворення, кожен із завданням установити ціну для максимізації прибутку, можуть навчитися змовлятися надзвичайно складним і цілком незаконним способом. Вони роблять це без явного навчання змові й без прямого спілкування: якимось чином їм удається організувати схему фіксації цін, лише спостерігаючи за цінами одне одного. Поки що це продемонстровано лише в моделюванні, а не в реальному ціноутворенні. Проте, за оцінками, значну частину цін в інтернеті встановлюють автономні ШІ, тому перспектива

масової фіксації цін непокоїть. Змова вигідна продавцям: якщо всі спільно встановлюють високі ціни, прибуток зростає, — але шкодить споживачам. Навіть не маючи такого наміру, продавці потенційно можуть використовувати ШІ для дій, які їм заборонено чинити прямо.[9] Це ще одне обличчя явища відмивання рішень математикою, яке я згадувала в розділі 7. Людям доведеться стежити, щоб зловмисники не ошукували їхні ШІ й щоб самі ШІ випадково не перетворювалися на зловмисників.

ЦЕ КРАЩЕ ДОРУЧИТИ ШІ

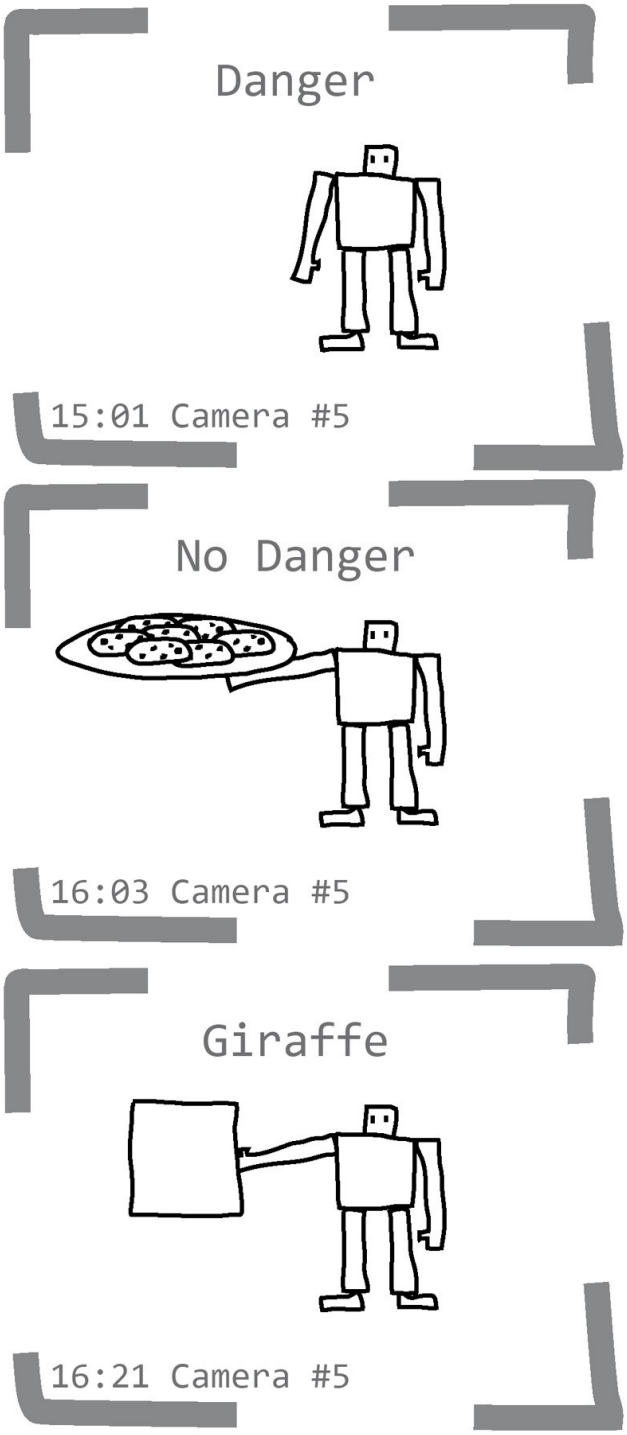
ШІ добре створює чернетки, а людина перетворює їх на готовий результат

Для багатьох алгоритмів машинного навчання робота на рівні людини є золотим стандартом. Зрештою, часто їхнє завдання — наслідувати приклади людських дій: підписування картинок, фільтрування електронної пошти, вигадування імен для морських свинок. Коли результат приблизно відповідає людському рівню, алгоритми можна — під наглядом — використовувати замість людей для повторюваних або нудних завдань. У попередніх розділах ми бачили, що деякі новинні організації застосовують машинне навчання для автоматичного створення нудних, але прийнятних статей про місцевий спорт або нерухомість. Проєкт Quicksilver автоматично створює чернетки статей *Вікіпедії* про жінок-науковиць, яких у *Вікіпедії* помітно бракує, заощаджуючи час редакторів-волонтерів. Люди, яким потрібно розшифровувати аудіо або перекладати тексти, використовують версії машинного навчання — хай і з помилками — як початкову чернетку для власної роботи. Музиканти можуть застосовувати алгоритми генерування музики, щоб створити оригінальну композицію точної тривалості для рекламного блоку, де музика має бути не винятковою, а просто дешевою. У багатьох випадках роль людини полягає в редагуванні.

Люди відвертіші з роботами, а робот із печивом легше проникає в гуртожиток

А для деяких завдань людей навіть краще не використовувати. Люди охочіше розповідають про свої емоції або потенційно стигматизовану інформацію, якщо вважають, що говорять із роботом, а не з людиною.[10,11] (З іншого боку, медичні чатботи можуть пропустити серйозну загрозу здоров’ю.)[12] Ботів також навчали переглядати тривожні зображення й позначати можливі злочини, хоча вони схильні плутати пустельні пейзажі з людською плоттю.[13] Навіть злочин роботові може бути скоїти легше, ніж людині. У 2016 році студентка

Гарварду Серена Бут створила робота для перевірки деяких теорій про те, чи не надто люди довіряють роботам.[14] Бут сконструювала простого дистанційно керованого робота, який під'їжджав до студентів і просив впустити його до гуртожитку, доступ до якого відкривала ключ-картка. За таких умов його впустили лише 19 відсотків людей; цікаво, що в групах студенти робили це трохи частіше. Однак коли той самий робот казав, що доставляє печиво, його впускали 76 відсотків людей.

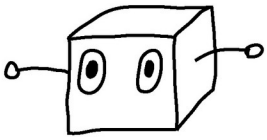


Складність рішень ШІ може приховувати незаконну дискримінацію

Як я згадувала вище, деякі ШІ можуть добре підходити й для злочинів через явище відмивання рішень математикою. Рішення ШІ здатні ґрунтуватися на складних зв’язках між кількома змінними, частина з яких може бути замінниками забороненої для нього інформації, наприклад гендеру чи раси. Це

додає шар непрозорості, який — навмисно чи ні — може дозволяти системі безкарно порушувати закони.

```
Task: ~~do crimes~~
play with the numbers and see what happens
```



ШІ швидший і послідовніший за людей, а після перевірок може бути й справедливішим

Є також багато випадків, коли ШІ кращий, бо перевершує людські можливості. По-перше, зазвичай він значно швидший за людей. У деяких багатокористувацьких комп’ютерних іграх ШІ доводиться навмисно сповільнювати, щоб люди мали хоча б якийсь шанс. ШІ також послідовніший, хоча жахливо справляється з несподіванками. Чи може ШІ бути ще й справедливішим? Потенційно так. Принаймні систему на основі ШІ можна перевірити на справедливість: виконати безліч тестових рішень і пошукати статистичні кореляції, яких там не повинно бути. Ретельно скоригувавши навчальні дані так, щоб їхня статистика відповідала світові, яким він *має бути*, а не світові, яким він є, у багатьох випадках можна навчити ШІ ухвалювати справедливі рішення — принаймні значно справедливіші, ніж рішення пересічної людини.

АЛГОРИТМІЧНА ТВОРЧІСТЬ?

Чи писатиме ШІ музику, фільми й романи майбутнього? Можливо, принаймні частково.

ШІ перетворює тиранозавра на квіти, а фортепіанне рифування — на гітарне соло

Створене ШІ мистецтво може бути вражаючим, дивним і тривожним: тюльпани, що нескінченно змінюють форму; глючні люди з напіврозплавленими обличчями; небеса, повні вигаданих собак. Тиранозавр може перетворитися на квіти або фрукти, *Мона Ліза* — розтягнути губи в дурнуватій усмішці, а фортепіанне рифування — перейти в соло електрогітари. Згенерований ШІ текст може набути якостей сюрреалістичного перформансу.

GPT-2 вигадує тварин із лускою, татуюваннями й обличчями жаб

> Отримавши підказку «My 10 favorite animals are: 1.» («Мої десять улюблених тварин: 1.»), нейромережа GPT-2 додала такий список:

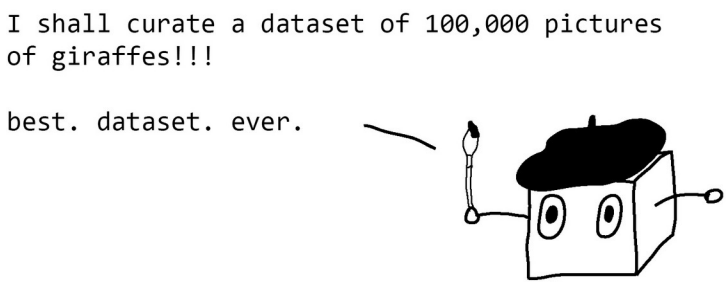
> *My 10 favorite animals are:* (Мої десять улюблених тварин:)
> *1. zebras with a white scar on the back;* (зебри з білим шрамом на спині;)
> *2. insiduous spiders and octopus;* (підступні павуки та восьминіг; оригінальне *insiduous* написане з помилкою)
> *3. frog with large leaves, hopefully black;* (жаба з великим листям, сподіваюся, чорним;)
> *4. cockatiel with scales;* (корела з лускою;)
> *5. razorbill with wings hanging about 4 inches from one's face and a heart tattoo on a frog.* (гагарка з крилами, що звисають приблизно за чотири дюйми від чийогось обличчя, і татуюванням серця на жабі.)

Творчість ШІ найточніше називати творчістю за допомогою ШІ

Як і розв'язання проблем за допомогою ШІ, творчість ШІ, мабуть, найкраще описувати як «творчість за допомогою ШІ».

Вибір і очищення набору даних уже є мистецьким рішенням

Щоб GAN створила картину, їй спершу потрібен набір даних, і саме людина вирішує, яким він буде. Деякі з найцікавіших результатів GAN з'являються, коли митці дають алгоритмам для навчання власні картини чи фотографії. Наприклад, художниця Анна Рідлер провела весну, зробивши десять тисяч фотографій тюльпанів, а потім використала їх для навчання GAN, яка створювала нескінченну низку майже фотореалістичних тюльпанів; смугастість кожної квітки була пов'язана з ціною біткоїна. Художниця та інженерка програмного забезпечення Гелена Сарін створила цікаві GAN-ремікси власних акварелей і ескізів, перетворивши їх на кубістичні або дивно текстуровані гібриди. Інші митці надихаються готовими наборами даних — наприклад, портретами доби Відродження чи пейзажами із суспільного надбання — і дивляться, що з них зробить GAN. Упорядкування набору даних теж є мистецьким актом. Додайте більше стилів живопису — і можете отримати гібридний або спотворений твір. Залиште лише один постійний ракурс, стиль або тип освітлення — і нейромережі буде легше відтворити побачене та створити реалістичніші зображення. Почніть із моделі, навченої на великому наборі, а потім за допомогою перенесення навчання зосередьте її на меншому, але спеціалізованішому наборі — й отримаєте ще більше способів точно налаштувати результат.



Письменник відкинув занадто шаблонну модель і залишив ту, що писала дивно

Люди, які навчають алгоритми генерувати текст, теж можуть керувати результатами через набори даних. Письменник-фантаст Робін Слоун — один із кількох авторів, які експериментують із текстом нейромереж як способом додати до письма трохи непередбачуваності.[15] Він створив власний інструмент, який відповідає на його речення, прогнозуючи наступне на основі знань з інших науково-фантастичних оповідань, наукових статей і навіть бюлетенів про охорону природи. Демонструючи інструмент в інтерв'ю *New York Times*, Слоун подав йому речення «Бізони зібралися навколо каньйону», а той продовжив: «під голим небом». Це не було ідеальним прогнозом: у реченні алгоритму відчувалося щось неправильне. Але для Слоуна воно було чудово дивним. Він навіть відкинув попередню модель, навчену на науковій фантастиці 1950-х і 1960-х років, бо її речення виявилися надто шаблонними.

Недонавчання, дивні збої й навіть колапс мод можуть стати художнім вибором

Як і збирання набору даних, навчання ШІ є мистецьким актом. Як довго воно має тривати? Не до кінця навчений ШІ інколи дає цікаві результати з дивними збоями або спотвореним правописом. Якщо ШІ застрягне й почне створювати нерозбірливий текст або дивні візуальні артефакти — наприклад, сітки, що множаться, чи перенасичені кольори; цей процес називають **колапсом мод**, — чи треба почати навчання спочатку? А може, цей ефект доволі крутий? Як і в інших застосуваннях, митець також повинен стежити, щоб ШІ не копіював вхідні дані надто точно. З погляду ШІ, точна копія набору даних — саме те, чого від нього хочуть, тому за першої нагоди він плагіатитиме.

Людина відбирає одну вдалу роботу з десятків або сотень посередніх результатів ШІ

І нарешті, саме художник-людина має відібрати результати ШІ та перетворити їх на щось варте уваги. GAN і алгоритми генерування тексту можуть створювати практично нескінченний обсяг матеріалу, але більшість із нього

нецікава. Дещо навіть жахливе: пам’ятайте, що багато текстових нейромереж не знають значення своїх слів. Так, я дивлюся на тебе, нейромереже, яка запропонувала назвати котів Mr. Tinkles і Retchion. Коли я навчаю нейромережі генерувати текст, показати варто лише крихітну частку результатів — одну десяту або одну соту. Я постійно відбираю результати, щоб представити історію чи цікаву думку про алгоритм або набір даних.

BigGAN змішує категорії, створюючи курособак і зелені м’ячі з носами

У деяких випадках відбір результатів ШІ може бути напрочуд складним процесом. У розділі 4 я використала BigGAN, щоб показати, як нейромережі для генерування зображень відчувають труднощі, коли навчаються на надто різноманітних картинках. Але я не розповіла про одну з найкрутіших її можливостей: створення зображень, що поєднують кілька категорій.

У прихованому просторі між собакою й тенісним м’ячем живуть пухнасті зелені сфери з носами

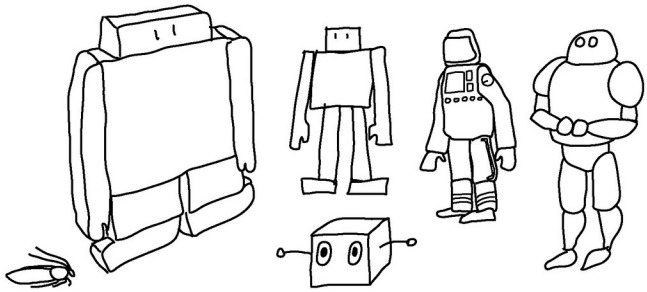
Уявіть «курку» як точку в просторі, а «собаку» — як іншу точку. Рухаючись найкоротшим шляхом між ними, ви проходите крізь інші точки, що перебувають десь посередині й де курособаки мають пір’я, обвислі вуха та висолоплені язики. Почніть із «собаки» й рухайтеся до «тенісного м’яча» — і пройдете крізь область пухнастих зелених сфер із чорними очима й носами, на які так і хочеться натиснути пальцем. Цей величезний багатовимірний візуальний ландшафт можливостей називається **прихованим простором**. Коли прихований простір BigGAN став доступним, митці почали занурюватися в нього й досліджувати. Вони швидко знайшли координати, де існували пальта, вкриті очима, і плащі, вкриті щупальцями; незграбні собакоптахи з обома очима на одному боці обличчя; бездоганні села гобітів із вишуканими круглими дверима; вогняні грибоподібні хмари з веселими мордочками цуценят. Як виявилося, в ImageNet дуже багато собак, тому прихований простір BigGAN теж переповнений собаками. Способи навігації прихованим простором самі стають мистецькими рішеннями. Рухатися прямими лініями чи кривими? Триматися поблизу початкової точки чи дозволити собі звернути в далекі крайні закутки? Кожен вибір кардинально змінює те, що ми побачимо. Досить утилітарні категорії ImageNet зливаються в цілковиту дивовижу.

ШІ генерує мистецтво, але творчу роботу все ще виконує людина

Чи все це мистецтво згенерував ШІ? Безумовно. Але чи саме ШІ виконав творчу роботу? І близько ні. Люди, які називають свої ШІ митцями,

перебільшують можливості ШІ — і водночас недооцінюють власний мистецький внесок та внесок людей, які спроєктували алгоритми.

ВИСНОВОК. Життя серед наших штучних друзів



ШІ дивує нас способами, яких не передбачають навіть його програмісти

На сторінках цієї книги ми побачили безліч різних способів, якими ШІ здатний нас дивувати.

ШІ падає вежею, натискає паузу й використовує збої фізики замість виконання задуму

Отримавши проблему й достатньо свободи у виборі способу її розв’язання, ШІ може вигадати рішення, про існування яких його програмісти навіть не мріяли. Доручіть йому пройти з точки А до точки Б — і він може натомість вирішити скласти себе у вежу та впасти. Або рухатиметься, обертаючись на місці тісними колами чи сіпаючись по підлозі звивистою купою. Якщо навчати його в симуляції, він може зламати саму тканину свого всесвіту, з’ясувавши, як використати збої фізики й набути надлюдських здібностей. Він сприйматиме вказівки буквально: коли йому скажуть уникати зіткнень, він відмовиться рухатися; коли скажуть не програвати відеогру, він знайде кнопку паузи й назавжди заморозить гру. Він знаходитиме закономірності, приховані в навчальних даних, навіть якщо програмісти їх не очікували. Деякі закономірності можуть бути такими, яких ми не хотіли б наслідувати, наприклад упередження. Модульні ШІ можуть каскадно поєднуватися й співпрацювати, щоб виконувати завдання, з якими не впорався б жоден окремий ШІ, діючи наче телефон, повний застосунків, або навіть бджолиний рій.

У найближчому майбутньому ШІ небезпечний не надмірним, а недостатнім розумом

Навіть стаючи дедалі здібнішим, ШІ все одно не знатиме, чого ми хочемо. Він усе одно *намагатиметься* зробити те, чого ми хочемо. Але між тим, що ми хочемо від ШІ, і тим, що наказуємо йому зробити, завжди може бути розрив. Чи стане він достатньо розумним, щоб розуміти нас і наш світ так само, як інша людина, — або навіть перевершити нас? Імовірно, не за нашого життя. У доступному для огляду майбутньому небезпека полягатиме не в тому, що ШІ надто розумний, а в тому, що він недостатньо розумний.

Фотореалістичні сцени й перемоги в іграх приховують звичайне зіставлення закономірностей

На поверхні здаватиметься, що ШІ розуміє дедалі більше. Він зможе генерувати фотореалістичні сцени, можливо, створювати цілі епізоди фільмів із розкішними текстурами, а можливо, перемагати в кожній комп’ютерній грі, яку ми йому запропонуємо. Але під усім цим лежить лише зіставлення закономірностей. Він знає тільки те, що бачив, причому бачив достатньо разів, щоб знайти в цьому сенс.

Світ надто дивний: ему втечуть, діти вдягнуть костюми тарганів, а люди запитають про жирафів

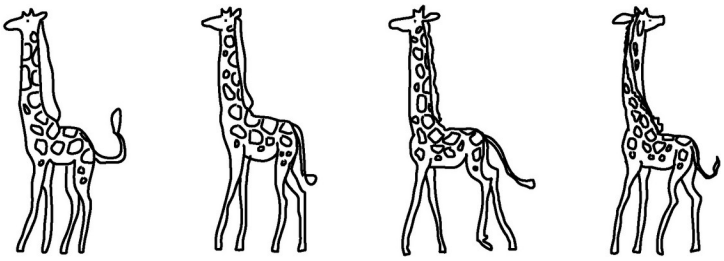
Наш світ надто складний, надто непередбачуваний і надто химерний, щоб ШІ побачив усе це під час навчання. Ему втечуть, діти почнуть носити костюми тарганів, а люди запитуватимуть про жирафів, навіть коли поруч немає жодного. ШІ неправильно розумітиме нас, бо йому бракуватиме контексту, щоб збагнути, чого ми насправді від нього хочемо.

Оптимізм щодо ШІ виправданий лише разом з обережністю та розумінням його обмежень

Щоб обрати найкращий шлях уперед разом із ШІ, нам доведеться розуміти його: знати, як вибирати правильні проблеми для розв’язання, як передбачати його непорозуміння та як не дозволити йому копіювати найгірше з того, що він знаходить у людських даних. У нас є всі причини оптимістично дивитися на ШІ — і всі причини бути обережними. Усе залежить від того, наскільки добре ми ним користуватимемося.

І остерігайтеся прихованих жирафів

І остерігайтеся тих прихованих жирафів.



Подяки

Цієї книги не існувало б без наполегливої праці, проникливості й щедрості багатьох людей, яким я з радістю дякую тут.

Величезна подяка команді Voracious, чия наполеглива праця перетворила мій розлогий і безладний документ на річ, яку я люблю. Літературне редагування Барбари Кларк незмірно поліпшило цю книгу, а вилучення цілої метричної тонни слів *«насправді»* зробило її легшою. Особлива подяка моїй редакторці Нікі Ґеррейро, яка одного дня несподівано написала мені електронного листа: це вже вп’яте їй довелося стримувати сміх у своєму офісі відкритого типу, тож чи не думала я про те, як мій блог можна перетворити на книгу? Без підтримки й гострої проникливості Нікі ця книга не мала б такого масштабу й сміливості.

Щиро дякую також моєму агентові Еріку Лупферу з Fletcher and Company, який життєрадісно провів авторку-дебютантку крізь численні етапи перетворення блогу на книгу.

Уперше я почула про машинне навчання у 2002 році, коли Ерік Ґудмен прочитав захопливу лекцію про еволюційні алгоритми для першокурсників Університету штату Мічиган. Мабуть, ті історії про алгоритми, які ламали симуляції та розв’язували не ту проблему, справді закарбувалися в моїй пам’яті! Дякую, що так рано запалили цю цікавість: вона подарувала мені стільки радості.

Дякую моїм друзям і родині, які підтримували мене впродовж цього довгого процесу, слухали мої пробні виступи й сміялися з моїх жартів, а також завжди були готові допомогти мені відновити сили за допомогою музики, походів або кулінарних експериментів.

І нарешті, дякую всім моїм читачам і підписникам на aiweirdness.com, які вже перетворили на реальність стільки моїх дивних експериментів зі ШІ: схеми для в’язання, печиво, лак для нігтів, бурлескні шоу, дивних істот, абсурдні котячі імена, назви пива й навіть оперу. Подивіться, що ми створили тепер! Нехай жирафи завжди будуть із вами.

Про авторку

Джанель Шейн має ступінь докторки філософії з електротехніки та магістерський ступінь із фізики. На сайті aiweirdness.com вона пише про штучний інтелект і про кумедні, а часом тривожні способи, якими алгоритми неправильно розуміють людські речі. Її включили до списку ста найкреативніших людей у бізнесі за версією *Fast Company*, а 2019 року вона стала спікеркою TED Talks. Її матеріали публікувалися в *New York Times, Slate, The New Yorker, The Atlantic, Popular Science* та інших виданнях. Вона майже напевно не робот.

Примітки

Вступ

1. Керолайн О’Донован та ін., «We Followed YouTube’s Recommendation Algorithm Down the Rabbit Hole» [«Ми пішли за рекомендаційним алгоритмом YouTube у кролячу нору»], *BuzzFeed News*, 24 січня 2019 року, https://www.buzzfeednews.com/article/carolineodonovan/down-youtubes-recommendation-rabbithole.

Розділ 1. Що таке ШІ?

1. Джоел Леман та ін., «The Surprising Creativity of Digital Evolution: A Collection of Anecdotes from the Evolutionary Computation and Artificial Life Research Communities» [«Дивовижна творчість цифрової еволюції: збірка історій від дослідницьких спільнот еволюційних обчислень і штучного життя»], arXiv:1803.03453 [cs], 9 березня 2018 року, http://arxiv.org/abs/1803.03453.
2. Ніл В. Патель, «Why Doctors Aren’t Afraid of Better, More Efficient AI Diagnosing Cancer» [«Чому лікарі не бояться кращого й ефективнішого ШІ для діагностування раку»], *The Daily Beast*, 11 грудня 2017 року, https://www.thedailybeast.com/why-doctors-arent-afraid-of-better-more-efficient-ai-diagnosing-cancer.
3. Джефф Ларсон та ін., «How We Analyzed the COMPAS Recidivism Algorithm» [«Як ми аналізували алгоритм оцінювання рецидивізму COMPAS»], *ProPublica*, 23 травня 2016 року, https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm.
4. Крис Вільямс, «AI Guru Ng: Fearing a Rise of Killer Robots Is Like Worrying about Overpopulation on Mars» [«Гуру ШІ Ин: боятися повстання роботів-убивць — це як непокоїтися через перенаселення Марса»], *The Register*, 19 березня 2015 року, https://www.theregister.co.uk/2015/03/19/andrew_ng_baidu_ai/.
5. Маріанна Бертран і Сендхіл Муллайнатан, «Are Emily and Greg More Employable Than Lakisha and Jamal? A Field Experiment on Labor Market Discrimination» [«Чи легше Емілі та Ґреґу знайти роботу, ніж Лакіші й Джамалу? Польовий експеримент щодо дискримінації на ринку праці»], *American Economic Review* 94, № 4 (вересень 2004): 991–1013, https://doi.org/10.1257/0002828042002561.

Розділ 2. ШІ всюди, але де саме?

1. Стівен Чень, «A Giant Farm in China Is Breeding 6 Billion Cockroaches a Year. Here’s Why» [«Гігантська ферма в Китаї вирощує шість мільярдів тарганів на рік. Ось навіщо»], *South China Morning Post*, 19 квітня 2018 року, https://www.scmp.com/news/china/society/article/2142316/giant-indoor-farm-china-breeding-six-billion-cockroaches-year.
2. Heliograf, «High School Football This Week: Einstein at Quince Orchard» [«Шкільний футбол цього тижня: Einstein проти Quince Orchard»], *Washington Post*, 13 жовтня 2017 року, https://www.washingtonpost.com/allmetsports/2017-fall/games/football/87408/.
3. Лі Л’Естрад, «MittMedia Homeowners Bot Boosts Digital Subscriptions with Automated Articles» [«Бот домовласників MittMedia збільшує кількість цифрових передплат завдяки автоматизованим статтям»], International News Media Association, 18 червня 2018 року, https://www.inma.org/blogs/ideas/post.cfm/mittmedia-homeowners-bot-boosts-digital-subscriptions-with-automated-articles.
4. Джаклін Пейзер, «The Rise of the Robot Reporter» [«Поява робота-репортера»], *New York Times*, 5 лютого 2019 року, https://www.nytimes.com/2019/02/05/business/media/artificial-intelligence-journalism-robots.html.

5. Крістофер Дж. Шаллу та Ендрю Вандербурґ, «Identifying Exoplanets with Deep Learning: A Five Planet Resonant Chain around Kepler-80 and an Eighth Planet around Kepler-90» [«Виявлення екзопланет за допомогою глибокого навчання»], *The Astronomical Journal* 155, № 2 (30 січня 2018): 94, https://doi.org/10.3847/1538-3881/aa9e09.

6. Р. Бентон Меткалф та ін., «The Strong Gravitational Lens Finding Challenge» [«Змагання з пошуку сильних гравітаційних лінз»], *Astronomy & Astrophysics* 625 (травень 2019): A119, https://doi.org/10.1051/0004-6361/201832797.

7. Аві Баґла, «#StarringJohnCho Level 2: Using DeepFakes for Representation» [«Використання дипфейків для репрезентації»], відео на YouTube, 9 квітня 2018 року, https://www.youtube.com/watch?v=hlZkATlqDSM&feature=youtu.be.

8. Том Саймоніт, «Facebook Built the Perfect Chatbot but Can’t Give It to You Yet» [«Facebook створив ідеального чатбота, але поки не може дати його вам»], *MIT Technology Review*, 14 квітня 2017 року, https://www.technologyreview.com/s/604117/facebooks-perfect-impossible-chatbot/.

9. Там само.

10. Кейсі Ньютон, «Facebook Is Shutting Down M, Its Personal Assistant Service That Combined Humans and AI» [«Facebook закриває M — сервіс персонального помічника, що поєднував людей і ШІ»], *The Verge*, 8 січня 2018 року, https://www.theverge.com/2018/1/8/16856654/facebook-m-shutdown-bots-ai.

11. Ендрю Дж. Гокінс, «Inside Waymo’s Strategy to Grow the Best Brains for Self-Driving Cars» [«Як Waymo вирощує найкращі “мізки” для безпілотних автомобілів»], *The Verge*, 9 травня 2018 року, https://www.theverge.com/2018/5/9/17307156/google-waymo-driverless-cars-deep-learning-neural-net-interview.

12. «OpenAI Five», OpenAI, дата звернення: 3 серпня 2019 року, https://openai.com/five/.

13. Катянна Куач, «OpenAI Bots Smashed in Their First Clash against Human Dota 2 Pros» [«Ботів OpenAI розгромили в першій сутичці з професійними гравцями Dota 2»], *The Register*, 23 серпня 2018 року, https://www.theregister.co.uk/2018/08/23/openai_bots_defeated/.

14. Том Мерфі (@tom7), Twitter, 23 серпня 2018 року, https://twitter.com/tom7/status/1032756005107580929.

15. Майк Кук (@mtrc), Twitter, 23 серпня 2018 року, https://twitter.com/mtrc/status/1032783369254432773.

16. Том Мерфі, «The First Level of Super Mario Bros. Is Easy with Lexicographic Orderings and Time Travel… After That It Gets a Little Tricky» [«Перший рівень Super Mario Bros. легко пройти за допомогою лексикографічного впорядкування й подорожей у часі… далі стає трохи складніше»], дослідницька робота, Carnegie Mellon University, 1 квітня 2013 року, http://www.cs.cmu.edu/~tom7/mario/mario.pdf.

17. Беньямін Сольник та ін., «Bayesian Optimization for a Better Dessert» [«Баєсівська оптимізація для кращого десерту»], доповідь на NIPS Workshop on Bayesian Optimization, Лонґ-Біч, Каліфорнія, 9 грудня 2017 року, https://bayesopt.github.io/papers/2017/37.pdf.

18. Сара Кімморлі, «We Tasted the ‘Perfect’ Cookie Google Took 2 Months and 59 Batches to Create—and It Was Terrible» [«Ми скуштували “ідеальне” печиво, на яке Google витратив два місяці й 59 партій, — і воно було жахливим»], *Business Insider Australia*, 31 травня 2018 року, https://www.businessinsider.com.au/google-smart-cookie-ai-recipe-2018-5.

19. Ендрю Крок, «Waymo’s Self-Driving Cars Are Far from Perfect, Report Says» [«Безпілотні автомобілі Waymo далекі від досконалості»], *Roadshow*, 28 серпня 2018 року, https://www.cnet.com/roadshow/news/waymo-alleged-tech-troubles-report/.

20. К. Лв та ін., «Analysis of Autopilot Disengagements Occurring during Autonomous Vehicle Testing» [«Аналіз відключень автопілота під час випробувань автономних автомобілів»], *IEEE/CAA Journal of Automatica Sinica* 5, № 1 (січень 2018): 58–68, https://doi.org/10.1109/JAS.2017.7510745.

21. Ендрю Крок, «Uber Self-Driving Car Saw Pedestrian 6 Seconds before Crash, NTSB Says» [«Безпілотний автомобіль Uber побачив пішохода за шість секунд до аварії, повідомляє

NTSB»], *Roadshow*, 24 травня 2018 року, https://www.cnet.com/roadshow/news/uber-self-driving-car-ntsb-preliminary-report/.

22. Фред Ламберт, «Tesla Elaborates on Autopilot's Automatic Emergency Braking Capacity over Mobileye's System» [«Tesla пояснює можливості автоматичного аварійного гальмування автопілота порівняно із системою Mobileye»], *Electrek*, 2 липня 2016 року, https://electrek.co/2016/07/02/tesla-autopilot-mobileye-automatic-emergency-braking/.

23. Нааман Чжоу, «Volvo Admits Its Self-Driving Cars Are Confused by Kangaroos» [«Volvo визнає, що кенгуру збивають її безпілотні автомобілі з пантелику»], *The Guardian*, 1 липня 2017 року, https://www.theguardian.com/technology/2017/jul/01/volvo-admits-its-self-driving-cars-are-confused-by-kangaroos.

Розділ 3. Як він насправді навчається?

1. Ян Ґудфеллоу, Йошуа Бенджіо та Аарон Курвіль, *Deep Learning* [«Глибоке навчання»] (Кембридж, Массачусетс: MIT Press, 2016).

2. Шон Макґреґор та ін., «FlareNet: A Deep Learning Framework for Solar Phenomena Prediction» [«FlareNet: система глибокого навчання для передбачення сонячних явищ»], доповідь на 31-й Conference on Neural Information Processing Systems, Лонґ-Біч, Каліфорнія, 8 грудня 2017 року, https://dl4physicalsciences.github.io/files/nips_dlps_2017_5.pdf.

3. Алек Редфорд, Рафал Юзефович та Ілля Суцкевер, «Learning to Generate Reviews and Discovering Sentiment» [«Навчання генерувати відгуки й виявлення тональності»], ArXiv:1704.01444 [Cs], 5 квітня 2017 року, http://arxiv.org/abs/1704.01444.

4. Андрей Карпаті, «The Unreasonable Effectiveness of Recurrent Neural Networks» [«Нерозумно висока ефективність рекурентних нейронних мереж»], Andrej Karpathy Blog, 21 травня 2015 року, http://karpathy.github.io/2015/05/21/rnn-effectiveness/.

5. Кріс Ола та ін., «The Building Blocks of Interpretability» [«Складники інтерпретованості»], *Distill* 3, № 3 (6 березня 2018): e10, https://doi.org/10.23915/distill.00010.

6. Девід Бау та ін., «GAN Dissection: Visualizing and Understanding Generative Adversarial Networks» [«Розтин GAN: візуалізація та розуміння генеративно-змагальних мереж»], доповідь на International Conference on Learning Representations, 6–9 травня 2019 року, https://gandissect.csail.mit.edu/.

7. «Botnik Apps», Botnik, дата звернення: 3 серпня 2019 року, http://botnik.org/apps.

8. Періс Мартіно, «Why Google Docs Is Gaslighting Everyone about Spelling: An Investigation» [«Чому Google Docs змушує всіх сумніватися у правописі: розслідування»], *The Outline*, 7 травня 2018 року, https://theoutline.com/post/4437/why-google-docs-thinks-real-words-are-misspelled.

9. Шаокан Чжан та ін., «Zoonotic Source Attribution of *Salmonella enterica* Serotype Typhimurium Using Genomic Surveillance Data, United States» [«Визначення зоонозного джерела Salmonella enterica серотипу Typhimurium за даними геномного нагляду у США»], *Emerging Infectious Diseases* 25, № 1 (2019): 82–91, https://doi.org/10.3201/eid2501.180835.

10. Ян Дж. Ґудфеллоу та ін., «Generative Adversarial Networks» [«Генеративно-змагальні мережі»], ArXiv:1406.2661 [Cs, Stat], 10 червня 2014 року, http://arxiv.org/abs/1406.2661.

11. Ахмед Ельґаммаль та ін., «CAN: Creative Adversarial Networks, Generating 'Art' by Learning About Styles and Deviating from Style Norms» [«CAN: творчі змагальні мережі, що створюють "мистецтво", вивчаючи стилі та відхиляючись від стильових норм»], ArXiv:1706.07068 [Cs], 21 червня 2017 року, http://arxiv.org/abs/1706.07068.

12. Беккет Мафсон, «This Artist Is Teaching Neural Networks to Make Abstract Art» [«Цей митець навчає нейронні мережі створювати абстрактне мистецтво»], *Vice*, 22 травня 2016 року, https://www.vice.com/en_us/article/yp59mg/neural-network-abstract-machine-paintings.

13. Девід Га та Юрґен Шмідгубер, «World Models» [«Моделі світу»], Zenodo, 28 березня 2018 року, https://doi.org/10.5281/zenodo.1207631.

Розділ 4. Він старається!

1. Теро Каррас, Самулі Лайне та Тімо Айла, «A Style-Based Generator Architecture for Generative Adversarial Networks» [«Архітектура генератора на основі стилю для генеративно-змагальних мереж»], ArXiv:1812.04948 [Cs, Stat], 12 грудня 2018 року, http://arxiv.org/abs/1812.04948.

2. Емілі Дрейфусс, «A Bot Panic Hits Amazon Mechanical Turk» [«Паніка через ботів охопила Amazon Mechanical Turk»], *Wired*, 17 серпня 2018 року, https://www.wired.com/story/amazon-mechanical-turk-bot-panic/.

3. «COCO Dataset», COCO: Common Objects in Context, http://cocodataset.org/#download. Під час навчання використали навчальні й перевірні зображення 2014 року — загалом 124 тисячі. Кожен діалог мав 10 запитань. За даними https://visualdialog.org/data, у навчальному наборі було 364 мільйони діалогів, тож кожне зображення зустрічалося 364 / 1,24 = 293,5 раза.

4. Гокінс, «Inside Waymo's Strategy» [«Усередині стратегії Waymo»].

5. Теро Каррас та ін., «Progressive Growing of GANs for Improved Quality, Stability, and Variation» [«Поступове нарощування GAN для поліпшення якості, стабільності та різноманітності»], ArXiv:1710.10196 [Cs, Stat], 27 жовтня 2017 року, http://arxiv.org/abs/1710.10196.

6. Каррас, Лайне та Айла, «A Style-Based Generator Architecture» [«Архітектура генератора на основі стилю»].

7. Мелісса Елліотт (0xabad1dea), «How Math Can Be Racist: Giraffing» [«Як математика може бути расистською: жирафлення»], Tumblr, 31 січня 2019 року, https://abad1dea.tumblr.com/post/182455506350/how-math-can-be-racist-giraffing.

8. Корінн Пертилл і Зої Шланґер, «Wikipedia Rejected an Entry on a Nobel Prize Winner Because She Wasn't Famous Enough» [«Wikipedia відхилила статтю про майбутню нобелівську лауреатку, бо вона була недостатньо відома»], *Quartz*, 2 жовтня 2018 року, https://qz.com/1410909/wikipedia-had-rejected-nobel-prize-winner-donna-strickland-because-she-wasnt-famous-enough/.

9. Джон Крістіан, «Why Is Google Translate Spitting Out Sinister Religious Prophecies?» [«Чому Google Translate видає зловісні релігійні пророцтва?»], *Vice*, 20 липня 2018 року, https://www.vice.com/en_us/article/j5npeg/why-is-google-translate-spitting-out-sinister-religious-prophecies.

10. Ніколас Карліні та ін., «The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks» [«Таємний оповідач: оцінювання й тестування ненавмисного запам'ятовування в нейронних мережах»], ArXiv:1802.08232 [Cs], 22 лютого 2018 року, http://arxiv.org/abs/1802.08232.

11. Йонас Йонґеян та ін., «Quick, Draw! The Data» [«Quick, Draw! Дані»] (набір для онлайн-гри Quick, Draw!), дата звернення: 3 серпня 2019 року, https://quickdraw.withgoogle.com/data.

12. Джон Енґлсман (@engelsjk), Google AI Quickdraw Visualizer (вебдемо), Github, дата звернення: 3 серпня 2019 року, https://engelsjk.github.io/web-demo-quickdraw-visualizer/.

13. Ґретхен Маккаллок, «Autocomplete Presents the Best Version of You» [«Автодоповнення показує найкращу версію вас»], *Wired*, 11 лютого 2019 року, https://www.wired.com/story/autocomplete-presents-the-best-version-of-you/.

14. Абгішек Дас та ін., «Visual Dialog» [«Візуальний діалог»], ArXiv:1611.08669 [Cs], 26 листопада 2016 року, http://arxiv.org/abs/1611.08669.

Розділ 5. Чого ви насправді просите?

1. @citizen_of_now, Twitter, 15 березня 2018 року, https://twitter.com/citizen_of_now/status/974344339815129089.

2. Даґ Бланк (@DougBlank), Twitter, 13 квітня 2018 року, https://twitter.com/DougBlank/status/984811881050329099.

3. @Smingleigh, Twitter, 7 листопада 2018 року, https://twitter.com/Smingleigh/status/1060325665671692288.

4. Крістін Баррон, «Pass the Butter // Pancake Bot» [«Передайте масло // МлинцеБот»], Unity Connect, січень 2018 року, https://connect.unity.com/p/pancake-bot.

5. Алекс Ірпан, «Deep Reinforcement Learning Doesn't Work Yet» [«Глибоке навчання з підкріпленням поки не працює»], Sorta Insightful (блог), 14 лютого 2018 року, https://www.alexirpan.com/2018/02/14/rl-hard.html.

6. Стерлінґ Кріспін (@sterlingcrispin), Twitter, 16 квітня 2018 року, https://twitter.com/sterlingcrispin/status/985967636302327808.

7. Сара Чодош, «The Problem with Cancer-Sniffing Dogs» [«Проблема із собаками, які винюхують рак»], 4 жовтня 2016 року, *Popular Science*, https://www.popsci.com/problem-with-cancer-sniffing-dogs/.

8. Wikipedia, стаття «Anti-Tank Dog» [«Протитанковий собака»], останнє оновлення — 29 червня 2019 року, https://en.wikipedia.org/w/index.php?title=Anti-tank_dog&oldid=904053260.

9. Анушка де Роган, «Why Dolphins Are Deep Thinkers» [«Чому дельфіни — глибокі мислителі»], *The Guardian*, 3 липня 2003 року, https://www.theguardian.com/science/2003/jul/03/research.science.

10. Сандіп Джаухар, «When Doctor's Slam the Door» [«Коли лікарі зачиняють двері»], *New York Times Magazine*, 16 березня 2003 року, https://www.nytimes.com/2003/03/16/magazine/when-doctor-s-slam-the-door.html.

11. Джоел Рубін (@joelrubin), Twitter, 6 грудня 2017 року, https://twitter.com/joelrubin/status/938574971852304384.

12. Джоел Саймон, «Evolving Floorplans» [«Еволюційне створення планів приміщень»], joelsimon.net, дата звернення: 3 серпня 2019 року, http://www.joelsimon.net/evo_floorplans.html.

13. Мерфі, «First Level of Super Mario Bros.» [«Перший рівень Super Mario Bros.»].

14. Том Мерфі (suckerpinch), «Computer Program that Learns to Play Classic NES Games» [«Комп'ютерна програма, яка вчиться грати в класичні ігри NES»], відео на YouTube, опубліковано 1 квітня 2013 року, https://www.youtube.com/watch?v=xOCurBYI_gY.

15. Мерфі, «First Level of Super Mario Bros.» [«Перший рівень Super Mario Bros.»].

16. Джек Кларк і Даріо Амодей, «Faulty Reward Functions in the Wild» [«Хибні функції винагороди в реальних умовах»], OpenAI, 22 грудня 2016 року, https://openai.com/blog/faulty-reward-functions/.

17. Bitmob, «Dimming the Radiant AI in Oblivion» [«Приборкання Radiant AI в Oblivion»], *VentureBeat* (блог), 17 грудня 2010 року, https://venturebeat.com/2010/12/17/dimming-the-radiant-ai-in-oblivion/.

18. cliffracer333, «So what happened to Oblivion's npc "goal" system that they used in the beta of the game. Is there a mod or a way to enable it again?» [«То що сталося із системою "цілей" NPC в Oblivion, яку використовували в бета-версії гри? Чи є мод або спосіб увімкнути її знову?»], гілка на Reddit, 10 червня 2016 року, https://www.reddit.com/r/oblivion/comments/4nimvh/so_what_happened_to_oblivions_npc_goal_system/.

19. Сіндія Н. Бхану, «A Desert Spider with Astonishing Moves» [«Пустельний павук із дивовижними рухами»], *New York Times*, 4 травня 2014 року, https://www.nytimes.com/2014/05/06/science/a-desert-spider-with-astonishing-moves.html.

20. Леман та ін., «The Surprising Creativity of Digital Evolution» [«Дивовижна творчість цифрової еволюції»].

21. Єтте Рандлев і Пребен Альстрем, «Learning to Drive a Bicycle Using Reinforcement Learning and Shaping» [«Навчання їзди на велосипеді за допомогою навчання з підкріпленням і формування поведінки»], *Proceedings of the Fifteenth International Conference on Machine Learning, ICML '98* (Сан-Франциско, Каліфорнія: Morgan Kaufmann Publishers Inc., 1998), 463–471, http://dl.acm.org/citation.cfm?id=645527.757766.

22. Юваль Тасса та ін., «DeepMind Control Suite» [«Набір засобів керування DeepMind»], ArXiv:1801.00690 [Cs], 2 січня 2018 року, http://arxiv.org/abs/1801.00690.

23. Бенджамін Рехт, «Clues for Which I Search and Choose» [«Підказки, які я шукаю й обираю»], блог arg min, 20 березня 2018 року, http://benjamin-recht.github.io/2018/03/20/mujocoloco/.

24. @citizen_of_now, Twitter, 15 березня 2018 року, https://twitter.com/citizen_of_now/status/974344339815129089.

25. Вестлі Ваймер, «Advances in Automated Program Repair and a Call to Arms» [«Досягнення в автоматизованому виправленні програм і заклик до дії»], *Search Based Software Engineering*, ред. Ґюнтер Руе і Юаньюань Чжан (Берлін і Гайдельберґ: Springer, 2013), 1–3.

26. Леман та ін., «The Surprising Creativity of Digital Evolution» [«Дивовижна творчість цифрової еволюції»].

27. Юрій Бурда та ін., «Large-Scale Study of Curiosity-Driven Learning» [«Масштабне дослідження навчання, керованого допитливістю»], ArXiv:1808.04355 [Cs, Stat], 13 серпня 2018 року, http://arxiv.org/abs/1808.04355.

28. А. Баранес і П.-І. Удейє, «R-IAC: Robust Intrinsically Motivated Exploration and Active Learning» [«R-IAC: надійне внутрішньо мотивоване дослідження й активне навчання»], *IEEE Transactions on Autonomous Mental Development* 1, № 3 (жовтень 2009 року): 155–169, https://doi.org/10.1109/TAMD.2009.2037513.

29. Девін Колдвей, «This Clever AI Hid Data from Its Creators to Cheat at Its Appointed Task» [«Цей кмітливий ШІ приховав дані від своїх творців, щоб схитрувати у призначеному завданні»], *TechCrunch*, 31 грудня 2018 року, http://social.techcrunch.com/2018/12/31/this-clever-ai-hid-data-from-its-creators-to-cheat-at-its-appointed-task/.

30. «YouTube Now: Why We Focus on Watch Time» [«YouTube зараз: чому ми зосереджуємося на часі перегляду»], YouTube Creator Blog, 10 серпня 2012 року, https://youtube-creators.googleblog.com/2012/08/youtube-now-why-we-focus-on-watch-time.html.

31. Ґійом Шасло (@gchaslot), Twitter, 9 лютого 2019 року, https://twitter.com/gchaslot/status/1094359568052817920?s=21.

32. «Continuing Our Work to Improve Recommendations on YouTube» [«Продовжуємо вдосконалювати рекомендації YouTube»], Official YouTube Blog, 25 січня 2019 року, https://youtube.googleblog.com/2019/01/continuing-our-work-to-improve.html.

Розділ 6. Злам Матриці, або ШІ знаходить спосіб

1. Даґ Бланк (@DougBlank), Twitter, 15 березня 2018 року, https://twitter.com/DougBlank/status/974244645214588930.

2. Нік Стеннінґ (@nickstenning), Twitter, 9 квітня 2018 року, https://twitter.com/DougBlank/status/974244645214588930.

3. Крістіан Ґаньє та ін., «Human-Competitive Lens System Design with Evolution Strategies» [«Проєктування системи лінз, конкурентоспроможної з людськими розробками, за допомогою еволюційних стратегій»], *Applied Soft Computing* 8, № 4 (1 вересня 2008 року): 1439–1452, https://doi.org/10.1016/j.asoc.2007.10.018.

4. Леман та ін., «The Surprising Creativity of Digital Evolution» [«Дивовижна творчість цифрової еволюції»].

5. Карл Сімс, «Evolving 3D Morphology and Behavior by Competition» [«Еволюційне формування тривимірної морфології та поведінки через конкуренцію»], *Artificial Life* 1, № 4 (1 липня 1994 року): 353–372, https://doi.org/10.1162/artl.1994.1.4.353.

6. Карл Сімс, «Evolving Virtual Creatures» [«Еволюційне створення віртуальних істот»], *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’94* (Нью-Йорк: ACM, 1994), 15–22, https://doi.org/10.1145/192161.192167.

7. Леман та ін., «The Surprising Creativity of Digital Evolution» [«Дивовижна творчість цифрової еволюції»].

8. Девід Клементс (@davecl42), Twitter, 18 березня 2018 року, https://twitter.com/davecl42/status/975406071182479361.

9. Нік Чейні та ін., «Unshackling Evolution: Evolving Soft Robots with Multiple Materials and a Powerful Generative Encoding» [«Звільнення еволюції: еволюційне створення м'яких роботів із багатьох матеріалів і потужним генеративним кодуванням»], *ACM SIGEVOlution* 7, № 1 (серпень 2014 року): 11–23, https://doi.org/10.1145/2661735.2661737.

10. Джон Тіммер, «Meet Wolbachia: The Male-Killing, Gender-Bending, Gonad-Eating Bacteria» [«Знайомтеся: вольбахія — бактерія, яка вбиває самців, змінює стать і поїдає статеві залози»], *Ars Technica*, 24 жовтня 2011 року, https://arstechnica.com/science/news/2011/10/meet-wolbachia-the-male-killing-gender-bending-gonad-chomping-bacteria.ars.

11. @forgek_, Twitter, 10 жовтня 2018 року, https://twitter.com/forgek_/status/1050045261563813888.

12. Р. Фельдт, «Generating Diverse Software Versions with Genetic Programming: An Experimental Study» [«Створення різноманітних версій програмного забезпечення за допомогою генетичного програмування: експериментальне дослідження»], *IEE Proceedings—Software* 145, № 6 (грудень 1998 року): 228–236, https://doi.org/10.1049/ip-sen:19982444.

13. Джордж Джонсон, «Eurisko, the Computer With a Mind of Its Own» [«Eurisko — комп'ютер із власним розумом»], Alicia Patterson Foundation, оновлено 6 квітня 2011 року, https://aliciapatterson.org/stories/eurisko-computer-mind-its-own.

14. Ерік Шульте, Стефані Форрест і Вестлі Ваймер, «Automated Program Repair through the Evolution of Assembly Code» [«Автоматизоване виправлення програм через еволюцію асемблерного коду»], *Proceedings of the IEEE/ACM International Conference on Automated Software Engineering, ASE '10* (Нью-Йорк: ACM, 2010), 313–316, https://doi.org/10.1145/1858996.1859059.

Розділ 7. Небезпечні короткі шляхи

1. Марко Туліо Рібейру, Самір Сінґх і Карлос Ґестрін, «"Why Should I Trust You?": Explaining the Predictions of Any Classifier» [«"Чому я маю вам довіряти?": пояснення прогнозів будь-якого класифікатора»], ArXiv:1602.04938 [Cs, Stat], 16 лютого 2016 року, http://arxiv.org/abs/1602.04938.

2. Люк Окден-Рейнер, «Exploring the ChestXray14 Dataset: Problems» [«Дослідження набору ChestXray14: проблеми»], блог Luke Oakden-Rayner, 18 грудня 2017 року, https://lukeoakdenrayner.wordpress.com/2017/12/18/the-chestxray14-dataset-problems/.

3. Девід М. Лейзер та ін., «The Parable of Google Flu: Traps in Big Data Analysis» [«Притча про Google Flu: пастки аналізу великих даних»], *Science* 343, № 6176 (14 березня 2014 року): 1203–1205, https://doi.org/10.1126/science.1248506.

4. Ґіді Шпербер, «What I've Learned from Kaggle's Fisheries Competition» [«Чого я навчився з конкурсу Kaggle про рибальство»], *Medium*, 1 травня 2017 року, https://medium.com/@gidishperber/what-ive-learned-from-kaggle-s-fisheries-competition-92342f9ca779.

5. Дж. Берд і П. Лейзелл, «The Evolved Radio and Its Implications for Modelling the Evolution of Novel Sensors» [«Еволюційно створене радіо та його значення для моделювання еволюції нових сенсорів»], *Proceedings of the 2002 Congress on Evolutionary Computation, CEC'02 (Cat. No.02TH8600)*, т. 2 (2002 World Congress on Computational Intelligence—WCCI'02, Гонолулу, Гаваї, США: IEEE, 2002): 1836–1841, https://doi.org/10.1109/CEC.2002.1004522.

6. Ганна Фрай, *Hello World: Being Human in the Age of Algorithms* [*«Привіт, світе: як бути людиною в добу алгоритмів»*] (Нью-Йорк: W. W. Norton & Company, 2018).

7. Ло Бенішу, «The Web's Most Toxic Trolls Live in… Vermont?» [«Найтоксичніші тролі інтернету живуть у… Вермонті?»], *Wired*, 22 серпня 2017 року, https://www.wired.com/2017/08/internet-troll-map/.

8. Вайолет Блу, «Google's Comment-Ranking System Will Be a Hit with the Alt-Right» [«Система Google для ранжування коментарів сподобається альтернативним правим»], *Engadget*, 1 вересня 2017 року, https://www.engadget.com/2017/09/01/google-perspective-comment-ranking-system/.

9. Джессамін Вест (@jessamyn), Twitter, 24 серпня 2017 року, https://twitter.com/jessamyn/status/900867154412699649.

10. Робін Спір, «ConceptNet Numberbatch 17.04: Better, Less-Stereotyped Word Vectors» [«ConceptNet Numberbatch 17.04: кращі й менш стереотипні вектори слів»], блог ConceptNet, 24 квітня 2017 року, http://blog.conceptnet.io/posts/2017/conceptnet-numberbatch-17-04-better-less-stereotyped-word-vectors/.

11. Айлин Каліскан, Джоанна Дж. Брайсон і Арвінд Нараянан, «Semantics Derived Automatically from Language Corpora Contain Human-like Biases» [«Семантика, автоматично отримана з мовних корпусів, містить упередження, подібні до людських»], *Science* 356, № 6334 (14 квітня 2017 року): 183–186, https://doi.org/10.1126/science.aal4230.

12. Ентоні Ґ. Ґрінвальд, Деббі Е. Макґі та Джордан Л. К. Шварц, «Measuring Individual Differences in Implicit Cognition: The Implicit Association Test» [«Вимірювання індивідуальних відмінностей у неявному пізнанні: тест неявних асоціацій»], *Journal of Personality and Social Psychology* 74 (червень 1998 року): 1464–1480.

13. Браян А. Нозек, Махзарін Р. Банаджі та Ентоні Ґ. Ґрінвальд, «Math = Male, Me = Female, Therefore Math Not = Me» [«Математика = чоловіче, я = жіноче, отже, математика ≠ я»], *Journal of Personality and Social Psychology* 83, № 1 (липень 2002 року): 44–59.

14. Спір, «ConceptNet Numberbatch 17.04».

15. Ларсон та ін., «How We Analyzed the COMPAS» [«Як ми аналізували COMPAS»].

16. Джефф Ларсон і Джулія Енґвін, «Bias in Criminal Risk Scores Is Mathematically Inevitable, Researchers Say» [«Упередження в оцінках ризику злочинності математично неминуче, кажуть дослідники»], *ProPublica*, 30 грудня 2016 року, https://www.propublica.org/article/bias-in-criminal-risk-scores-is-mathematically-inevitable-researchers-say.

17. Джеймс Регальбуто, «Insurance Circular Letter No. 1 (2019)» [«Циркулярний лист про страхування № 1 (2019)»], Департамент фінансових послуг штату Нью-Йорк, 18 січня 2019 року, https://www.dfs.ny.gov/industry_guidance/circular_letters/cl2019_01.

18. Джеффрі Дастін, «Amazon Scraps Secret AI Recruiting Tool That Showed Bias against Women» [«Amazon відмовляється від таємного інструмента ШІ для найму, який виявив упередження проти жінок»], Reuters, 10 жовтня 2018 року, https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G.

19. Джеймс Вінсент, «Amazon Reportedly Scraps Internal AI Recruiting Tool That Was Biased against Women» [«За повідомленнями, Amazon відмовляється від внутрішнього інструмента ШІ для найму, упередженого проти жінок»], *The Verge*, 10 жовтня 2018 року, https://www.theverge.com/2018/10/10/17958784/ai-recruiting-tool-bias-amazon-report.

20. Паола Чеккі-Дімельйо, «How Gender Bias Corrupts Performance Reviews, and What to Do About It» [«Як гендерне упередження спотворює оцінки результативності та що із цим робити»], *Harvard Business Review*, 12 квітня 2017 року, https://hbr.org/2017/04/how-gender-bias-corrupts-performance-reviews-and-what-to-do-about-it.

21. Дейв Ґершгорн, «Companies Are on the Hook If Their Hiring Algorithms Are Biased» [«Компанії відповідають за упередженість своїх алгоритмів найму»], *Quartz*, 22 жовтня 2018 року, https://qz.com/1427621/companies-are-on-the-hook-if-their-hiring-algorithms-are-biased/.

22. Карен Хао, «Police across the US Are Training Crime-Predicting AIs on Falsified Data» [«Поліція по всій території США навчає ШІ для прогнозування злочинів на сфальсифікованих даних»], *MIT Technology Review*, 13 лютого 2019 року, https://www.technologyreview.com/s/612957/predictive-policing-algorithms-ai-crime-dirty-data/.

23. Стів Лор, «Facial Recognition Is Accurate, If You’re a White Guy» [«Розпізнавання облич точне, якщо ви білий чоловік»], *New York Times*, 9 лютого 2018 року, https://www.nytimes.com/2018/02/09/technology/facial-recognition-race-artificial-intelligence.html.

24. Джулія Карпентер, «Google’s Algorithm Shows Prestigious Job Ads to Men, but Not to Women. Here’s Why That Should Worry You» [«Алгоритм Google показує чоловікам оголошення про престижні посади, а жінкам — ні. Ось чому це має вас турбувати»], *Washington Post*, 6 липня

2015 року, https://www.washingtonpost.com/news/the-intersect/wp/2015/07/06/googles-algorithm-shows-prestigious-job-ads-to-men-but-not-to-women-heres-why-that-should-worry-you/.

25. Марк Вілсон, «This Breakthrough Tool Detects Racism and Sexism in Software» [«Цей проривний інструмент виявляє расизм і сексизм у програмному забезпеченні»], *Fast Company*, 22 серпня 2017 року, https://www.fastcompany.com/90137322/is-your-software-secretly-racist-this-new-tool-can-tell.

26. ORCAA, дата звернення: 3 серпня 2019 року, http://www.oneilrisk.com.

27. Фейсал Каміран і Тон Калдерс, «Data Preprocessing Techniques for Classification without Discrimination» [«Методи попередньої обробки даних для класифікації без дискримінації»], *Knowledge and Information Systems* 33, № 1 (1 жовтня 2012 року): 1–33, https://doi.org/10.1007/s10115-011-0463-8.

Розділ 8. Чи схожий мозок ШІ на людський?

1. Га і Шмідгубер, «World Models» [«Моделі світу»].

2. Ентоні Дж. Белл і Терренс Дж. Сейновскі, «The “Independent Components” of Natural Scenes Are Edge Filters» [«“Незалежні компоненти” природних сцен є фільтрами країв»], *Vision Research* 37, № 23 (1 грудня 1997 року): 3327–3338, https://doi.org/10.1016/S0042-6989(97)00121-1.

3. Андреа Баніно та ін., «Vector-Based Navigation Using Grid-Like Representations in Artificial Agents» [«Векторна навігація за допомогою ґратчастих представлень у штучних агентів»], *Nature* 557, № 7705 (травень 2018 року): 429–433, https://doi.org/10.1038/s41586-018-0102-6.

4. Бау та ін., «GAN Dissection» [«Розтин GAN»].

5. Ларрі С. Єґер, «Computational Genetics, Physiology, Metabolism, Neural Systems, Learning, Vision, and Behavior or PolyWorld: Life in a New Context» [«Обчислювальна генетика, фізіологія, метаболізм, нервові системи, навчання, зір і поведінка, або PolyWorld: життя в новому контексті»], *Santa Fe Institute Studies in the Sciences of Complexity*, т. 17 (Лос-Аламос, Нью-Мексико: Addison-Wesley Publishing Company, 1994), 262–263.

6. Баба Нарумі та ін., «Trophic Eggs Compensate for Poor Offspring Feeding Capacity in a Subsocial Burrower Bug» [«Трофічні яйця компенсують слабку здатність годувати потомство в субсоціального рийного клопа»], *Biology Letters* 7, № 2 (23 квітня 2011 року): 194–196, https://doi.org/10.1098/rsbl.2010.0707.

7. Роберт М. Френч, «Catastrophic Forgetting in Connectionist Networks» [«Катастрофічне забування в конекціоністських мережах»], *Trends in Cognitive Sciences* 3, № 4 (квітень 1999 року): 128–135.

8. Цзеюй Чжао та ін., «Men Also Like Shopping: Reducing Gender Bias Amplification Using Corpus-Level Constraints» [«Чоловіки теж люблять ходити по крамницях: зменшення посилення гендерного упередження за допомогою обмежень на рівні корпусу»], ArXiv:1707.09457 [Cs, Stat], 28 липня 2017 року, http://arxiv.org/abs/1707.09457.

9. Денні Кармон, Деніел Зоран і Йоав Ґолдберґ, «LaVAN: Localized and Visible Adversarial Noise» [«LaVAN: локалізований і видимий змагальний шум»], ArXiv:1801.02608 [Cs], 8 січня 2018 року, http://arxiv.org/abs/1801.02608.

10. Ендрю Ільяс та ін., «Black-Box Adversarial Attacks with Limited Queries and Information» [«Змагальні атаки на чорну скриньку з обмеженими запитами та інформацією»], ArXiv:1804.08598 [Cs, Stat], 23 квітня 2018 року, http://arxiv.org/abs/1804.08598.

11. Баттіста Біґджо та ін., «Poisoning Behavioral Malware Clustering» [«Отруєння поведінкового кластерування шкідливих програм»], ArXiv:1811.09985 [Cs, Stat], 25 листопада 2018 року, http://arxiv.org/abs/1811.09985.

12. Том Вайт, «Synthetic Abstractions» [«Синтетичні абстракції»], *Medium*, 23 серпня 2018 року, https://medium.com/@tom_25234/synthetic-abstractions-8f0e8f69f390.

13. Семюел Ґ. Фінлейсон та ін., «Adversarial Attacks Against Medical Deep Learning Systems» [«Змагальні атаки проти медичних систем глибокого навчання»], ArXiv:1804.05296 [Cs, Stat], 14 квітня 2018 року, http://arxiv.org/abs/1804.05296.

14. Філіп Бонтрейґер та ін., «DeepMasterPrints: Generating MasterPrints for Dictionary Attacks via Latent Variable Evolution» [«DeepMasterPrints: створення майстер-відбитків для словникових атак через еволюцію прихованих змінних»], ArXiv:1705.07386 [Cs], 20 травня 2017 року, http://arxiv.org/abs/1705.07386.

15. Стівен Бураньї, «How to Persuade a Robot That You Should Get the Job» [«Як переконати робота, що роботу слід отримати саме вам»], *The Observer*, 4 березня 2018 року, https://www.theguardian.com/technology/2018/mar/04/robots-screen-candidates-for-jobs-artificial-intelligence.

16. Лорен Джонсон, «4 Deceptive Mobile Ad Tricks and What Marketers Can Learn From Them» [«Чотири оманливі трюки мобільної реклами й чого маркетологи можуть із них навчитися»], *Adweek*, 16 лютого 2018 року, https://www.adweek.com/digital/4-deceptive-mobile-ad-tricks-and-what-marketers-can-learn-from-them/.

17. Віланд Брендель і Маттіас Бетґе, «Approximating CNNs with Bag-of-Local-Features Models Works Surprisingly Well on ImageNet» [«Наближення CNN моделями мішка локальних ознак напрочуд добре працює на ImageNet»], ArXiv:1904.00760 [Cs, Stat], 20 березня 2019 року, http://arxiv.org/abs/1904.00760.

Розділ 9. Люди-боти (де вам не варто очікувати ШІ?)

1. @yoco68, Twitter, 12 липня 2018 року, https://twitter.com/yoco68/status/1017404857190268928.

2. Пармі Олсон, «Nearly Half of All “AI Startups” Are Cashing in on Hype» [«Майже половина всіх “ШІ-стартапів” заробляє на ажіотажі»], *Forbes*, 4 березня 2019 року, https://www.forbes.com/sites/parmyolson/2019/03/04/nearly-half-of-all-ai-startups-are-cashing-in-on-hype/#5b1c4a66d022.

3. Керолін Саїд, «Kiwibots Win Fans at UC Berkeley as They Deliver Fast Food at Slow Speeds» [«Kiwibots завойовують прихильників у Каліфорнійському університеті в Берклі, повільно доставляючи фастфуд»], *San Francisco Chronicle*, 26 травня 2019 року, https://www.sfchronicle.com/business/article/Kiwibots-win-fans-at-UC-Berkeley-as-they-deliver-13895867.php.

4. Олівія Солон, «The Rise of “Pseudo-AI”: How Tech Firms Quietly Use Humans to Do Bots’ Work» [«Поширення “псевдо-ШІ”: як технологічні компанії потай використовують людей для виконання роботи ботів»], *The Guardian*, 6 липня 2018 року, https://www.theguardian.com/technology/2018/jul/06/artificial-intelligence-ai-humans-bots-tech-companies.

5. Еллен Г’юет, «The Humans Hiding Behind the Chatbots» [«Люди, які ховаються за чатботами»], *Bloomberg.com*, 18 квітня 2016 року, https://www.bloomberg.com/news/articles/2016-04-18/the-humans-hiding-behind-the-chatbots.

6. Річард Рей, «SpinVox Answers BBC Allegations over Use of Humans Rather than Machines» [«SpinVox відповідає на звинувачення BBC у використанні людей замість машин»], *The Guardian*, 23 липня 2009 року, https://www.theguardian.com/business/2009/jul/23/spinvox-answer-back.

7. Бекі Лер (@Breakaribecca), Twitter, 7 липня 2018 року, https://twitter.com/Breakaribecca/status/1015787072102289408.

8. Пол Мозур, «Inside China’s Dystopian Dreams: A.I., Shame and Lots of Cameras» [«Усередині антиутопічних мрій Китаю: ШІ, сором і безліч камер»], *New York Times*, 8 липня 2018 року, https://www.nytimes.com/2018/07/08/business/china-surveillance-technology.html.

9. Аарон Маміт, «Facebook AI Invents Language That Humans Can’t Understand: System Shut Down Before It Evolves Into Skynet» [«ШІ Facebook винайшов мову, якої люди не розуміють: систему вимкнули, перш ніж вона еволюціонувала у Skynet»], *Tech Times*, 30 липня 2017 року, http://www.techtimes.com/articles/212124/20170730/facebook-ai-invents-language-that-humans-cant-understand-system-shut-down-before-it-evolves-into-skynet.htm.

10. Кайл Біґґерс, «Babysitter Screening App Predictim Uses AI to Sniff out Bullies» [«Застосунок Predictim для перевірки нянь використовує ШІ, щоб винюхувати кривдників»], *VentureBeat*

(блог), 4 жовтня 2018 року, https://venturebeat.com/2018/10/04/babysitter-screening-app-predictim-uses-ai-to-sniff-out-bullies/.

11. Челсі Ґод, «Here's What Sophia, the First Robot Citizen, Thinks About Gender and Consciousness» [«Ось що Софія, перший робот-громадянин, думає про гендер і свідомість»], *Live Science*, 11 липня 2018 року, https://www.livescience.com/63023-sophia-robot-citizen-talks-gender.html.

12. К. Д. Мартін, «ENIAC: Press Conference That Shook the World» [«ENIAC: пресконференція, яка приголомшила світ»], *IEEE Technology and Society Magazine* 14, № 4 (зима 1995 року): 3–10, https://doi.org/10.1109/44.476631.

13. Александра Петрі, «A Bot Named "Eugene Goostman" Passes the Turing Test… Kind Of» [«Бот на ім'я "Юджин Ґустман" проходить тест Тюрінга… певною мірою»], *Washington Post*, 9 червня 2014 року, https://www.washingtonpost.com/blogs/compost/wp/2014/06/09/a-bot-named-eugene-goostman-passes-the-turing-test-kind-of/.

14. Браян Мерчант, «Predictim Claims Its AI Can Flag "Risky" Babysitters. So I Tried It on the People Who Watch My Kids» [«Predictim стверджує, що його ШІ здатний виявляти "ризикованих" нянь. Тож я випробував його на людях, які доглядають моїх дітей»], *Gizmodo*, 6 грудня 2018 року, https://gizmodo.com/predictim-claims-its-ai-can-flag-risky-babysitters-so-1830913997.

15. Дрю Гарвелл, «AI Start-up That Scanned Babysitters Halts Launch Following Post Report» [«ШІ-стартап, який перевіряв нянь, зупинив запуск після матеріалу Post»], *Washington Post*, 14 грудня 2018 року, https://www.washingtonpost.com/technology/2018/12/14/ai-start-up-that-scanned-babysitters-halts-launch-following-post-report/.

16. Тоня Райлі, «Get Ready, This Year Your Next Job Interview May Be with an A.I. Robot» [«Готуйтеся: цього року вашу наступну співбесіду може проводити робот зі ШІ»], CNBC, 13 березня 2018 року, https://www.cnbc.com/2018/03/13/ai-job-recruiting-tools-offered-by-hirevue-mya-other-start-ups.html.

17. Там само.

Розділ 10. Партнерство людини та ШІ

1. Ту Нгуєн-Фуок та ін., «HoloGAN: Unsupervised Learning of 3D Representations from Natural Images» [«HoloGAN: неконтрольоване навчання тривимірних представлень на природних зображеннях»], ArXiv:1904.01326 [Cs], 2 квітня 2019 року, http://arxiv.org/abs/1904.01326.

2. Дрю Лінслі та ін., «Learning What and Where to Attend» [«Навчання того, на що й де звертати увагу»], ArXiv:1805.08819 [Cs], 22 травня 2018 року, http://arxiv.org/abs/1805.08819.

3. Гектор Ї (@eigenhector), Twitter, 14 вересня 2018 року, https://twitter.com/eigenhector/status/1040501195989831680.

4. Вілл Найт, «A Tougher Turing Test Shows That Computers Still Have Virtually No Common Sense» [«Складніший тест Тюрінга показує, що комп'ютери й досі практично не мають здорового глузду»], *MIT Technology Review*, 14 липня 2016 року, https://www.technologyreview.com/s/601897/tougher-turing-test-exposes-chatbots-stupidity/.

5. Джеймс Регалбуто, «Insurance Circular Letter» [«Інформаційний лист про страхування»].

6. Еббі Олгайзер, «Trolls Turned Tay, Microsoft's Fun Millennial AI Bot, into a Genocidal Maniac» [«Тролі перетворили Tay, веселого ШІ-бота Microsoft для міленіалів, на маніяка-геноцидника»], *Chicago Tribune*, 26 березня 2016 року, https://www.chicagotribune.com/business/ct-internet-breaks-microsoft-ai-bot-tay-20160326-story.html.

7. Ґлен Леві, «Google's Bizarre Search Helper Assumes We Have Parakeets, Diarrhea» [«Дивний помічник пошуку Google припускає, що в нас є папужки й діарея»], *Time*, 4 листопада 2010 року, http://newsfeed.time.com/2010/11/04/why-why-wont-my-parakeet-eat-my-diarrhea-is-on-google-trends/.

8. Майкл Айзен, «Amazon's $23,698,655.93 Book about Flies» [«Книга Amazon про мух за 23 698 655,93 долара»], *It Is NOT Junk* (блог), 22 квітня 2011 року, http://www.michaeleisen.org/blog/?p=358.

9. Еміліо Кальвано та ін., «Artificial Intelligence, Algorithmic Pricing, and Collusion» [«Штучний інтелект, алгоритмічне ціноутворення та змова»], *VoxEU* (блог), 3 лютого 2019 року, https://voxeu.org/article/artificial-intelligence-algorithmic-pricing-and-collusion.

10. Солон, «The Rise of “Pseudo-AI”» [«Поширення “псевдо-ШІ”»].

11. Ґейл М. Лукас та ін., «It’s Only a Computer: Virtual Humans Increase Willingness to Disclose» [«Це лише комп’ютер: віртуальні люди збільшують готовність ділитися особистим»], *Computers in Human Behavior* 37 (1 серпня 2014 року): 94–100, https://doi.org/10.1016/j.chb.2014.04.043.

12. Ліліана Ларанджо та ін., «Conversational Agents in Healthcare: A Systematic Review» [«Розмовні агенти в охороні здоров’я: систематичний огляд»], *Journal of the American Medical Informatics Association* 25, № 9 (1 вересня 2018 року): 1248–1258, https://doi.org/10.1093/jamia/ocy072.

13. Марґі Мерфі, «Artificial Intelligence Will Detect Child Abuse Images to Save Police from Trauma» [«Штучний інтелект виявлятиме зображення жорстокого поводження з дітьми, щоб уберегти поліцію від травми»], *The Telegraph*, 18 грудня 2017 року, https://www.telegraph.co.uk/technology/2017/12/18/artificial-intelligence-will-detect-child-abuse-images-save/.

14. Адам Зеве, «In Automaton We Trust» [«На автомата покладаємося»], Гарвардська школа інженерії та прикладних наук, 25 травня 2016 року, https://www.seas.harvard.edu/news/2016/05/in-automaton-we-trust.

15. Девід Стрейтфельд, «Computer Stories: A.I. Is Beginning to Assist Novelists» [«Комп’ютерні історії: ШІ починає допомагати романістам»], *New York Times*, 18 жовтня 2018 року, https://www.nytimes.com/2018/10/18/technology/ai-is-beginning-to-assist-novelists.html.

Виноски

* Давня приказка про те, що мавпа, яка нескінченно довго випадково натискатиме клавіші друкарської машинки, зрештою надрукує всі твори Шекспіра, насправді досить точно описує метод «грубої сили»: пошук розв’язання через систематичне випробування всіх можливостей. В ідеалі використання ШІ має бути кращим за цей підхід. В ідеалі.

* Те, що на цьому етапі рахунок становить 27:0, а не 28:0, означає: Cougars, можливо, не реалізували одне додаткове очко після тачдауну. Heliograf цього не згадує.

* Мережа також мала крихітну довгострокову пам’ять, у якій могла зберігати інформацію довше, ніж у вікні із шістдесяти п’яти символів, але її було недостатньо для цілого переліку інгредієнтів. У термінах машинного навчання це робить алгоритм нейромережею **довгої короткочасної пам’яті (LSTM)**, а не звичайною RNN.

* У наборі категорію було написано *deserts* («пустелі»), а не *desserts* («десерти»), тому нейромережа вважає правильним саме таке написання.

* Алгоритм Google Translate постійно оновлюється, тому з часом ці результати істотно змінюватимуться.